

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

RESTAURACE AUDIOSIGNÁLŮ ZALOŽENÁ NA ŘÍDKÝCH
REPREZENTACÍCH

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. PAVEL ZÁVIŠKA

Brno 2017



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA ELEKTROTECHNIKY
A KOMUNIKAČNÍCH TECHNOLOGIÍ**

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

**RESTAURACE AUDIOSIGNÁLŮ ZALOŽENÁ NA ŘÍDKÝCH
REPREZENTACÍCH**

AUDIO RESTORATION BASED ON SPARSE SIGNAL REPRESENTATIONS

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Pavel Záviška

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Mgr. Pavel Rajmic, Ph.D.

BRNO 2017

Diplomová práce

magisterský navazující studijní obor **Telekomunikační a informační technika**

Ústav telekomunikací

Student: Bc. Pavel Záviška

ID: 154913

Ročník: 2

Akademický rok: 2016/17

NÁZEV TÉMATU:

Restaurace audiosignálů založená na řídkých reprezentacích

POKYNY PRO VYPRACOVÁNÍ:

Prostudujte modely znehodnocení audiosignálu při omezení okamžité hodnoty signálu (clipping). Analyzujte tento problém a seznámte se s obvyklými metodami pro jeho řešení, jejich vlastnostmi. Seznamte se rovněž s moderními metodami založenými na tzv. řídkosti signálů a navrhnete základní metodu restaurace a její pokročilé modifikace. Algoritmy implementujte v MATLABu a účinnost metod otestujte a porovnejte na několika odlišných typech audiosignálů. Dále vyhodnoťte úspěšnost vhodným objektivním kritériem i subjektivně.

DOPORUČENÁ LITERATURA:

[1] Hrbáček, R., Rajmic, P., Veselý, V., Špiřík, J. Řídké reprezentace signálů: úvod do problematiky, Elektrevue, 2011. ISSN 1213-1539

[2] Špiřík, J., Rajmic, P., Veselý, V. Reprezentace signálů: od bází k framům, Elektrevue, 2010. ISSN 1213-1539

Termín zadání: 1.2.2017

Termín odevzdání: 24.5.2017

Vedoucí práce: doc. Mgr. Pavel Rajmic, Ph.D.

Konzultant:

doc. Ing. Jiří Mišurec, CSc.
předseda oborové rady

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Diplomová práce se zabývá problematikou audio clippingu a aplikací modelu řídkých reprezentací pro úlohu declippingu. Nejprve je popsána obecná teorie clippingu, na kterou navazuje stručný přehled stávajících metod a popis obecné teorie, zabývající se řídkými reprezentacemi signálů a bázemi, resp. framy. Následně jsou představeny dvě metody, které řeší úlohu declippingu na základě řídkých reprezentací. První metoda používá Obecný proximální algoritmus pro konvexní optimalizaci, druhá pak Douglas-Rachfordův algoritmus. Zmíněné metody byly naprogramovány v prostředí Matlab. Výsledky metod jsou vyhodnoceny podle ukazatelů SNR, PEMO-Q a také podle subjektivních poslechových testů.

KLÍČOVÁ SLOVA

clipping, declipping, audio, zpracování signálu, řídké reprezentace, řídkost, proximální dělení

ABSTRACT

This Master's Thesis deals with the issue of audio clipping and the application of sparse representations model for the task of declipping. First, a general theory of clipping is described, followed by a brief overview of existing methods and a description of the general theory concerning sparse representations of signals and bases, respectively frames. Subsequently, two methods solving declipping problem based on sparse representations are introduced. The first method uses the Generic proximal algorithm for convex optimization, the second one uses the Douglas-Rachford algorithm. The above mentioned methods have been programmed in the Matlab environment. The results of the declipping methods are evaluated according to SNR, PEMO-Q and also by subjective listening tests.

KEYWORDS

clipping, declipping, audio, signal processing, sparse representations, sparsity, proximal-splitting

ZÁVIŠKA, Pavel. *Restaurace audiosignálů založená na řídkých reprezentacích*. Brno, 2017, 94 s. Diplomová práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedoucí práce: doc. Mgr. Pavel Rajmic, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Restaurace audiosignálů založená na řídkých reprezentacích“ jsem vypracoval(a) samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor(ka) uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil(a) autorská práva třetích osob, zejména jsem nezasáhl(a) nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom(a) následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno

.....

podpis autora(-ky)

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu diplomové práce panu doc. Mgr. Pavlu Rajmicovi, Ph.D. za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Brno

.....

podpis autora(-ky)

PODĚKOVÁNÍ

Výzkum popsany v této diplomové práci byl realizován v laboratořích podpořených z projektu SIX; registrační číslo CZ.1.05/2.1.00/03.0072, operační program Výzkum a vývoj pro inovace.

Brno

.....
podpis autora(-ky)

OBSAH

Úvod	11
1 Clipping a jeho vliv na signál	12
1.1 Clipping	12
1.1.1 Hard clipping	13
1.1.2 Soft clipping	15
1.2 Clipping a aliasing	17
1.3 Declipping	17
2 Současné metody	19
3 Úvod do řídké reprezentace	28
3.1 Základní pojmy a značení	28
3.2 Syntetizující a analyzující model signálu	29
3.3 Hledání řídkého řešení	30
3.4 Algoritmy pro hledání řídkých řešení	32
4 Báze a framy	34
4.1 Vektorový prostor a generátory VP	34
4.2 Báze vektorového prostoru	34
4.3 Framy	35
4.4 Gaborova transformace a framy	35
4.5 Constant-Q transformace	37
5 Proximální algoritmy a jejich využití pro úlohu declippingu	42
5.1 Matematická formulace úlohy declipping	42
5.2 Proximální algoritmy	43
5.2.1 Proximální operátory	43
5.2.2 Algoritmy proximálního dělení	44
5.3 Řešení pomocí Condatova algoritmu	46
5.3.1 Declipping pomocí Condatova algoritmu	47
5.3.2 Projekce na množinu přípustných řešení	49
5.4 Řešení pomocí algoritmu Douglas-Rachford	50
5.4.1 Definice množiny C	50
5.4.2 Declipping pomocí algoritmu Douglas-Rachford	51
5.4.3 Projekce na množinu přípustných řešení	51

6	Programové řešení	53
6.1	Hlavní soubor	53
6.2	Parametry okna pro DGT	56
6.3	Clipping	56
6.4	Condatův algoritmus	57
6.5	Douglas-Rachfordův algoritmus	58
6.6	Segmentace	58
6.7	Projekce	58
7	Experimenty a výsledky	60
7.1	Testování	60
7.2	Srovnání proximálních algoritmů	61
7.3	Porovnání pomocí ukazatele SNR	63
7.3.1	DGT	63
7.3.2	CQT banka filtrů	64
7.3.3	Srovnání s konkurenčními metodami	68
7.4	Porovnání pomocí PEMO-Q	70
7.4.1	Saturované signály	70
7.4.2	DGT	71
7.4.3	CQT banka filtrů	72
7.4.4	Srovnání s konkurenčními metodami	74
7.5	Porovnání výpočetní náročnosti algoritmů	76
7.6	Poslechové testy	77
7.6.1	Metodika testování	77
7.6.2	Výsledky poslechových testů	78
8	Závěr	81
	Literatura	83
	Seznam symbolů, veličin a zkratk	88
	Seznam příloh	90
A	Grafy rekonstruovaných signálů	91
B	Obsah přiloženého CD	93
B.1	Sparse_Declip	93
B.2	Sounds	93
B.3	Testing	94

SEZNAM OBRÁZKŮ

1.1	Převodní charakteristika hard clippingu.	13
1.2	Tvrdě saturovaný signál a porovnání spekter.	14
1.3	Převodní charakteristika soft clippingu.	15
1.4	Měkce saturovaný signál a porovnání spekter.	16
3.1	Ilustrace jednotkových koulí.	32
4.1	Ukázka několika atomů Gaborova framu s Hannovým oknem, zobrazena pouze reálná část.	36
4.2	Spektrogram zvukového signálu akustické kytary.	37
4.3	Modulové kmitočtové charakteristiky filtrů rovnoměrné (uniformní) CQT.	38
4.4	Modulové kmitočtové charakteristiky filtrů nerovnoměrné CQT s konstantními ℓ_2 -normami.	39
4.5	CQT spektrogram signálu z akustické kytary.	39
4.6	Závislost modulových kmitočtových charakteristik CQT banky filtrů na těsnosti framu.	40
5.1	Převodní charakteristika měkkého prahování.	44
7.1	Typický průběh ukazatele ΔSNR v závislosti na počtu iterací.	60
7.2	Poměr výpočetních časů pro Condatův algoritmus a Douglas-Rachfordův algoritmus v případě použití DGT.	61
7.3	Porovnání průměrných průběhů ΔSNR pro Condatův algoritmus a Douglas-Rachfordův algoritmus.	62
7.4	Hodnoty ukazatele ΔSNR pro signály se slovníkem DGT.	64
7.5	Modulová kmitočtová charakteristika neuniformní banky filtrů CQT.	65
7.6	Hodnoty ukazatele ΔSNR pro signály s použitím neuniformní CQT.	65
7.7	Modulová kmitočtová charakteristika uniformní banky filtrů CQT.	66
7.8	Hodnoty ukazatele ΔSNR pro signály s použitím uniformní CQT a váhování koeficientů.	66
7.9	Hodnoty ukazatele ΔSNR pro signály s použitím uniformní CQT, váhování koeficientů a potlačením vysokofrekvenčních složek.	67
7.10	Srovnání testovaných algoritmů podle hodnot ΔSNR	69
7.11	Hodnoty ukazatele PSM_t pro saturované signály.	71
7.12	Hodnoty ukazatele PSM_t pro signály se slovníkem DGT.	72
7.13	Hodnoty ukazatele PSM_t pro signály s použitím neuniformní CQT.	72
7.14	Hodnoty ukazatele PSM_t pro signály s použitím uniformní CQT a váhování koeficientů.	73
7.15	Hodnoty ukazatele PSM_t pro signály s použitím uniformní CQT, váhování koeficientů a potlačením vysokofrekvenčních složek.	74

7.16	Srovnání testovaných algoritmů podle hodnot PSM_t .	75
7.17	Srovnání výpočetní náročnosti jednotlivých algoritmů.	76
7.18	Screenshot aplikace pro poslechové hodnocení kvality rekonstrukce.	78
7.19	Výsledky poslechového testování pro silně zkreslené signály ($\theta_c = 0,1$).	79
7.20	Výsledky poslechového testování pro středně zkreslené signály ($\theta_c = 0,3$).	80
7.21	Výsledky poslechového testování pro lehce zkreslené signály ($\theta_c = 0,5$).	80
A.1	Ukázka rekonstrukce signálu se slovníkem DGT.	91
A.2	Ukázka rekonstrukce signálu s použitím neuniformní CQT.	91
A.3	Ukázka rekonstrukce signálu s použitím váhované uniformní CQT.	92
A.4	Ukázka rekonstrukce signálu s použitím váhované uniformní CQT a potlačením vysokofrekvenčních složek.	92

ÚVOD

Audio signály obecně jsou velmi náchylné k různým typům poškození a degradace. Mezi nejčastější typy poškození signálu patří především tzv. clipping, který může vzniknout při pořizování zvukové nahrávky, a to dokonce několika způsoby (malý dynamický rozsah mikrofону, přebuzení vstupního zesilovače, extrémní komprese signálu, atd.). Tento efekt se pak projevuje na poslech nepříjemným zkreslením.

S rozvojem moderních matematických metod však vznikají i metody pro rekonstrukci poškozených zvukových záznamů. Tyto metody lze pak s výhodou použít k rekonstrukci poškozených vzorků signálu namísto nutnosti např. zvukovou stopu nahrávat znovu. Velký krok vpřed v této oblasti učinily především metody na bázi řídkých reprezentací. A právě těmito metodám se věnuje tato diplomová práce.

V první kapitole je obecně charakterizován clipping. Ve druhé kapitole je následně představeno několik stávajících metod, řešících rekonstrukci saturovaných signálů – tedy declipping. Pozornost je věnována především metodám na bázi řídkých reprezentací.

Třetí kapitola obsahuje úvod do řídkých reprezentací a poskytuje teoretický základ pro pochopení metod declippingu, které budou představeny v kapitole 5.

Čtvrtá kapitola se zabývá teorií vektorových prostorů, bází a framů, a popisuje také diskrétní Gaborovu transformaci a Constant-Q transformaci, které budou dále použity pro účely declippingu.

Pátá kapitola se věnuje samotnému řešení declippingu. Nejprve je úloha declippingu matematicky formulována, dále jsou představeny algoritmy proximálního dělení, a následně je popsána metoda declippingu pomocí obecného proximálního algoritmu včetně popisu použitých projekcí. V další části je stávající metoda pomocí malého triku v projekci upravena na tvar, který lze řešit pomocí Douglas-Rachfordova algoritmu, což pomůže kvalitě rekonstrukce i rychlosti výpočtu. Závěr kapitoly tvoří popis nové projekce.

Šestá kapitola navazuje na předchozí a popisuje vlastní implementaci jednotlivých algoritmů, včetně pomocných funkcí, možností nastavení i důležitých proměnných.

Poslední, sedmá kapitola pak demonstruje výsledky dosažené pomocí představených metod a srovnává je s některými konkurenčními metodami. Pro porovnání jsou použity objektivní ukazatele SNR a PEMO-Q a také subjektivní poslechové testy.

1 CLIPPING A JEHO VLIV NA SIGNÁL

V této kapitole se budu věnovat především problematice audio clippingu. Popíšu zde, kde se s clippingem můžeme setkat, vysvětlím, proč vzniká a popíšu dva hlavní typy clippingu – tzv. hard clipping a soft clipping. Uvedu zde i jednotlivé příklady na reálných signálech, pro názornost jak v časové, tak i ve frekvenční oblasti. Ve stručnosti shrnu problém aliasingu v souvislosti s clippingem a v poslední části dále vysvětlím úlohu declippingu.

1.1 Clipping

Clipping je forma nelineárního zkreslení signálu, která nastane, pokud zpracovávaný signál disponuje větším dynamickým rozsahem, než je systém schopný zpracovat. Výstupní signál systému pak bude přichycen na prahové úrovni systému a ztratíme tak část informace původního signálu.

S clippingem se můžeme setkat všude tam, kde je signál digitalizován nebo jakýmkoli způsobem zpracováván za přítomnosti změny zesílení (*gainu*) a to jak v digitální, tak i analogové formě [1].

Při zpracování audio signálu dochází nejčastěji ke clippingu v zesilovači, kdy je zesilovač nucen k většímu zesílení, než je jeho napájení schopno dodat. V tom případě se zesilovač dostane do tzv. stavu saturace, kdy bude na výstupu dodávat maximální možná napětí, které zvládne – své napájecí napětí. Jakákoli úroveň signálu, která by přesáhla při daném zesílení maximální úroveň zesilovače, bude limitována právě na napájecí napětí zesilovače [2].

Při poslechu se saturovaný signál projeví většinou jako nepříjemné zkreslení nebo praskání. Ne vždy je však clipping nežádoucí efekt – záměrně se používá např. při zpracování signálu z elektrofonické kytary k získání typického zvuku zkreslené kytary (viz 1.1.2).

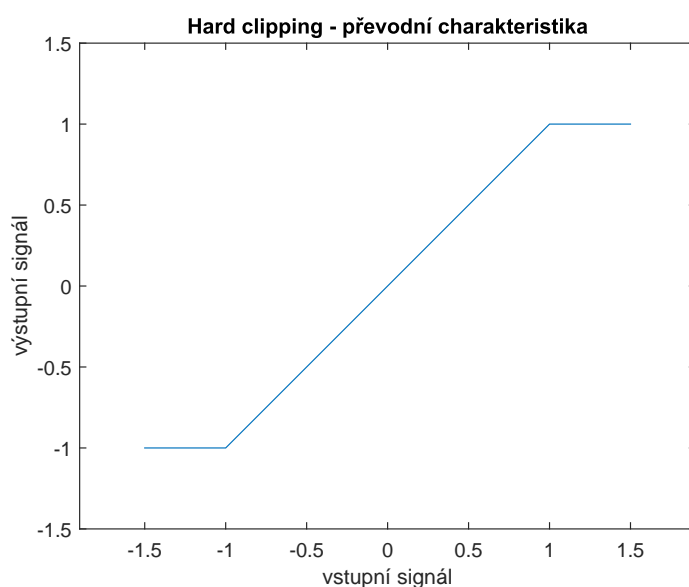
Ke clippingu pochopitelně nedochází jen při zpracování audio signálu, ale obecně jakéhokoliv signálu. Může se např. objevit i u fotografií, kdy limitaci způsobuje omezený dynamický rozsah světelného snímáče (CMOS, popř. CCD čipu). Na snímcích pak můžeme rozpoznat oblasti, které se jeví jako bílé (přexponované) nebo naopak černé (podexponované) [3].

1.1.1 Hard clipping

Pojmem hard clipping (popř. digitální clipping) máme na mysli tvrdé oříznutí signálu, pokud signál překročí úroveň θ_c . Matematicky to můžeme vyjádřit jako [4]:

$$y_c[n] = \begin{cases} y[n] & \text{pro } |y[n]| < \theta_c, \\ \theta_c \cdot \text{sgn}(y[n]) & \text{pro } |y[n]| \geq \theta_c. \end{cases} \quad (1.1)$$

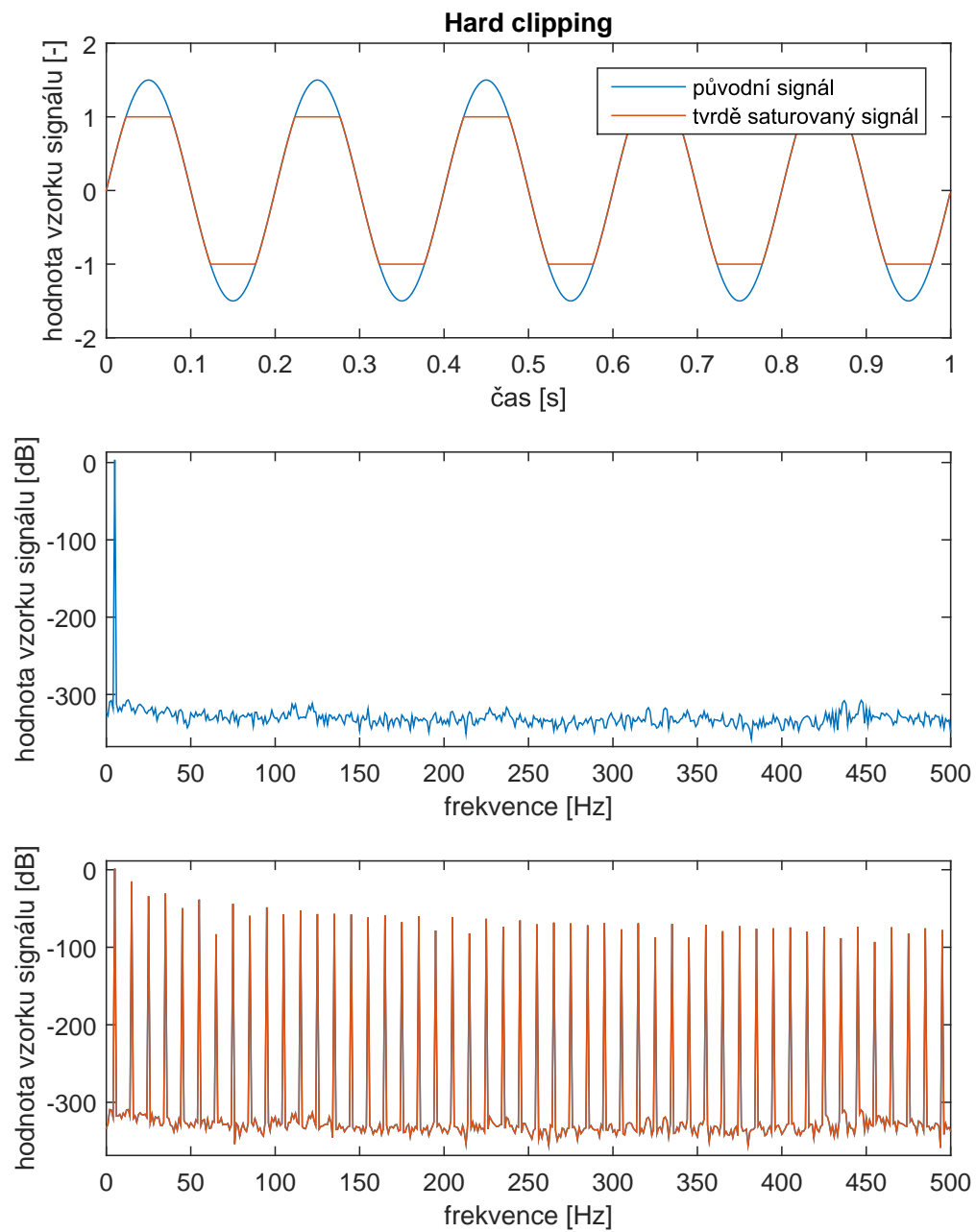
V tomto případě dochází k úplné ztrátě informace ve špičkách původního signálu a perfektní rekonstrukce saturovaného signálu je prakticky nemožná. Na obrázku 1.1 je zobrazena převodní charakteristika hard clippingu. Z ní je patrné, že mezi prahy $-\theta_c$ a θ_c je přenos ze vstupu na výstup lineární. Pokud ale vstupní signál překročí práh, bude výstupní signál limitován právě na hodnoty $\pm\theta_c$, v tomto případě ± 1 .



Obr. 1.1: Převodní charakteristika hard clippingu.

Na obrázku 1.2 pak můžeme vidět aplikaci hard clippingu na vstupní sinusový signál s kmitočtem 5 Hz a amplitudou 1,5 [-]. Tento signál byl pak zpracován systémem s přenosovou charakteristikou zobrazenou na obrázku 1.1.

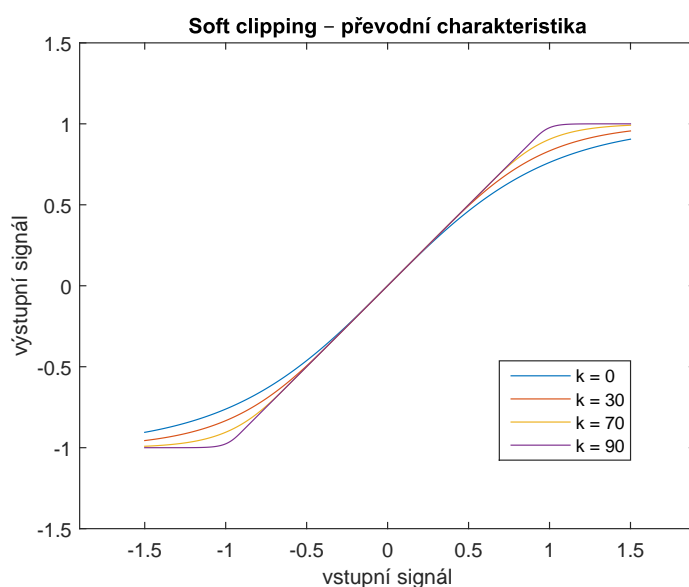
Zajímavé je také porovnání spekter původního a saturovaného signálu. Zatímco u původního signálu vidíme ve spektru pochopitelně pouze jednu složku na frekvenci 5 Hz, ve spektru saturovaného signálu je těchto složek teoreticky nekonečně mnoho. Tento jev je způsoben tím, že saturovaný signál už není hladká funkce a právě ostré hrany vyskytující se na přechodu k prahovým hodnotám $\pm\theta_c$ způsobují rozšíření spektra o liché harmonické složky původního signálu – tj. liché násobky původní frekvence. V našem případě se nám tedy ve spektru objeví složky na frekvencích 15 Hz, 25 Hz, 35 Hz, atd. Právě tyto liché harmonické složky ve spektru pak způsobují na poslech nepříjemné zkreslení saturovaného signálu.



Obr. 1.2: Tvrdě saturovaný signál a porovnání spekter.

1.1.2 Soft clipping

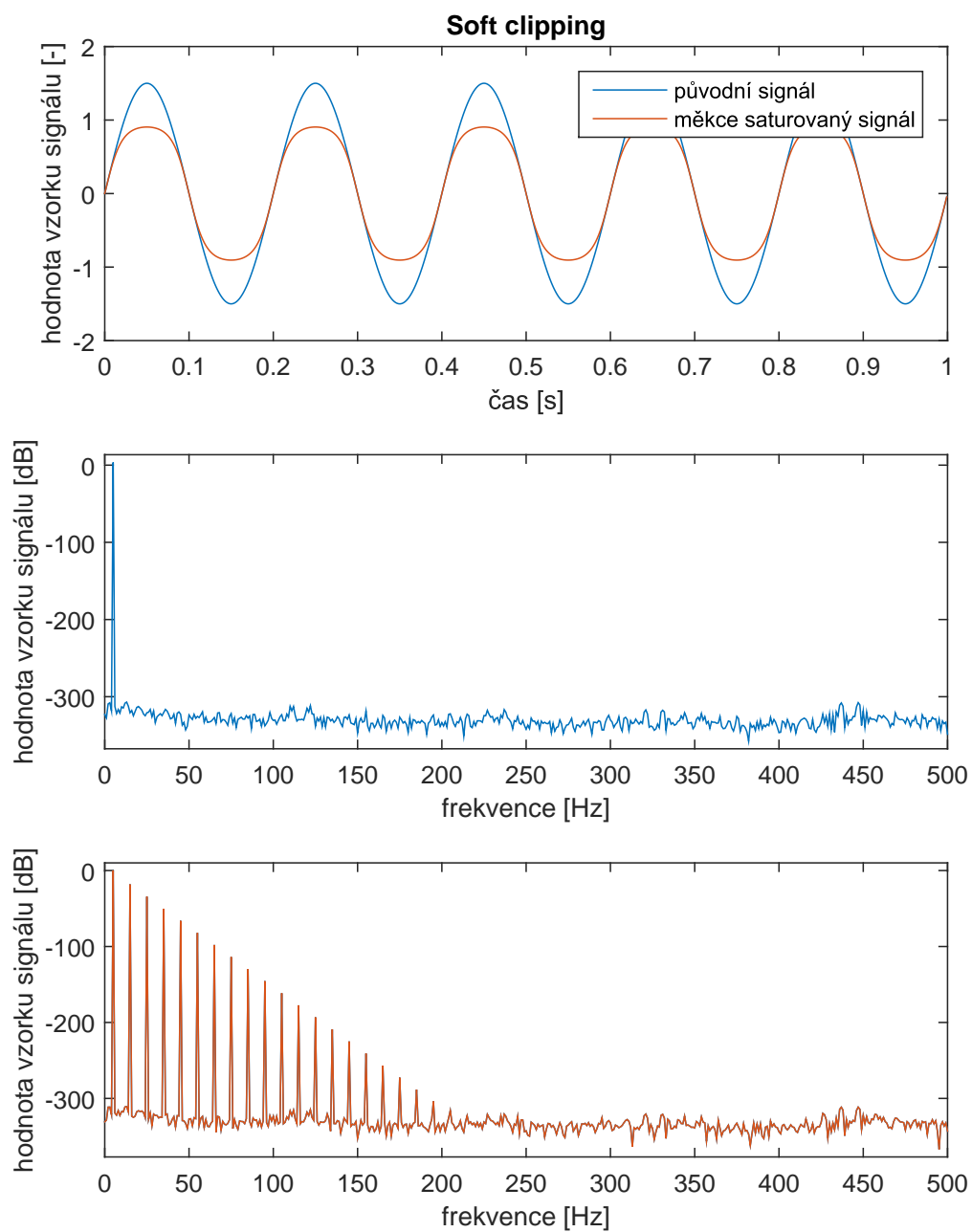
Soft clipping podobně jako hard clipping omezuje dynamiku vstupního signálu na hodnoty úrovní $\pm\theta_c$, nedochází zde však k tvrdému oříznutí špiček, ale k postupné kompresi špiček. Čím více se budeme blížit k úrovni θ_c , tím více bude signál komprimován, ale nikdy nebude tvrdě oříznut. Pokud tedy nebude úroveň zkreslení příliš velká, nedojde v žádném místě signálu k úplné ztrátě informace. Při znalosti převodní charakteristiky systému jsme dokonce schopni perfektně zrekonstruovat původní signál.



Obr. 1.3: Převodní charakteristika soft clippingu.

Jak už jsem zmínil dříve, soft clipping se většinou používá záměrně v kytarových zesilovačích a efektech typu overdrive, které vytvářejí typický charakter zkreslené elektrofonické kytary. Další příklad použití soft clippingu je v zesilovačích jako obrana před hard clippingem – zesilovač detekuje svoje výstupní napětí a pokud se začne blížit své prahové úrovni, začne své výstupní napětí komprimovat. Místo velkého zkreslení v podobě hard clippingu pak bude slyšet mírný pokles dynamiky signálu a jemné zkreslení, které ale nebude tak rušivé, jako v případě hard clippingu.

Převodní charakteristiku soft clippingu nelze jednoznačně definovat, protože záleží především na aplikaci a také na konkrétním výrobcí. Obecně však můžeme říci, že soft clipping lze modelovat libovolnou sigmoidou (např. $\arctan(x)$, $\tanh(x)$, $\frac{x}{1+|x|}$ nebo $\frac{x}{\sqrt{1+x^2}}$).



Obr. 1.4: Měkce saturovaný signál a porovnání spekter.

Pro svou aplikaci jsem zvolil funkci [5]:

$$y_c[n] = \begin{cases} y[n] & \text{pro } |y[n]| \leq r, \\ \text{sgn}(y[n]) \left[(\theta_c - r) \tanh\left(\frac{|y[n]| - r}{\theta_c - r}\right) + r \right] & \text{pro } |y[n]| > r, \end{cases} \quad (1.2)$$

kde θ_c je nastavitelný práh a r je práh, od kterého se začne výstupní signál komprimovat. To znamená, že mezi úrovněmi $-r$ a r je přenosová charakteristika lineární.

Pokud zavedeme proměnnou „tvrdost“ charakteristiky k , která bude nabývat hodnot od 0 do 100, pak r vypočítáme jako $r = 0,01 \cdot k \cdot \theta_c$. V tomto případě pak $k = 0$ bude nejměkčí možná charakteristika, naopak pro $k = 100$ bude $r = \theta_c$ a přenosová charakteristika bude stejná, jako v případě pro hard clipping. Na grafu 1.3 pak můžeme pozorovat převodní charakteristiku soft clippingu pro různé nastavení tvrdostí k .

Na obrázku 1.4 můžeme vidět aplikaci soft clippingu na sinusový vstupní signál. Opět je použit signál s kmitočtem 5 Hz a amplitudou 1,5 [-]. Pro soft clipping byla použita funkce (1.2) s úrovní $\theta_c = 1$ a koeficientem tvrdosti $k = 0$. Na průběhu signálu je patrné, jak soft clipping komprimuje špičky signálu, zatímco zbytek signálu zůstává nezměněn. Sinusoida po soft clippingu je pořád hladká funkce, což se projeví i v zobrazeném spektru – složky spektra signálu klesají mnohem rychleji než v případě hard clippingu.

1.2 Clipping a aliasing

Vlivem clippingu signálu může docházet i k negativnímu efektu zvanému aliasing. Je obecně známo, že aby nedocházelo k aliasingu, musí být splněna Nyquistova podmínka, která říká, že vzorkovací frekvence musí být minimálně dvakrát větší, než je nejvyšší harmonická složka obsažená ve vzorkovaném signálu.

Pokud budeme uvažovat clipping, při kterém vzniká teoreticky nekonečně mnoho harmonických složek, tyto složky se pak „ozrcadlí“ podle vzorkovacího kmitočtu a následně se sečtou s harmonickými složkami na nižších frekvencích. Díky tomu pak dojde k aliasingovému zkreslení signálu. V této práci však aliasing nebude uvažován.

1.3 Declipping

Úlohou declipping máme na mysli inverzní postup, kdy se ze saturovaného (tedy nějakým způsobem poškozeného) signálu snažíme získat zpět původní signál nebo alespoň takový signál, který by byl původnímu signálu co nejbližší a co nejlépe by byly odstraněny rušivé jevy zkreslení saturovaného signálu [6].

Ačkoli můžeme declipping aplikovat na oba typy clippingu (hard i soft), většina metod se zaměřuje právě na declipping saturovaného signálu po hard clippingu.

Jedním z důvodů je fakt, že negativní vliv hard clippingu na poslech je mnohem drastičtější než v případě soft clippingu. Navíc, pokud u soft clippingu známe převodní charakteristiku a míra zkreslení signálu se pohybuje v „rozumných“ mezích, pak je dokonce možné pomocí inverzní charakteristiky perfektně rekonstruovat původní signál.

2 SOUČASNÉ METODY

V této kapitole se dále budu věnovat stávajícím metodám řešícím problematiku declippingu a uvedu zde několik metod, kdy každá přistupuje k úloze jiným způsobem. Všechny metody však mají společnou jednu vlastnost – předpokládají, že je známo, u kterých vzorků signálu došlo vlivem clippingu k oříznutí. Jednotlivé články jsou zhruba seřazeny podle data vydání.

Adaptive interpolation of discrete-time signals that can be modeled as autoregressive processes

První článek, který nepřímo řeší problematiku declippingu je *Adaptive interpolation of discrete-time signals that can be modeled as autoregressive processes* [7], který vyšel už v roce 1986. Janssen a kolektiv v tomto článku představili algoritmus pro doplňování chybějících vzorků do signálů na základě iterativního výpočtu lineárních predikčních koeficientů a následné interpolace chybějících vzorků pomocí lineární predikce. Pokud budeme uvažovat saturované vzorky jako chybějící, lze pak tento algoritmus použít i pro declipping.

Restoring a Clipped Signal

První článek týkající se přímo problematiky declippingu (*Restoring a clipped signal* [8]) pochází z roku 1991 od autorů Abela a Smithe. Autoři v něm předpokládají, že zpracováváný signál má omezenou šířku pásma a je nadvzorkován. Nalezení signálu pak spočívá ve vyřešení relativně jednoduchého konvexního problému.

Statistical Model-Based Approaches to Audio Restoration and Analysis

Později Godsill a kol. použili v článku *Statistical model-based approaches to audio restoration and analysis* [9] parametrický model signálu a pomocí bayesovské statistiky určili chybějící vzorky jako ty nejpravděpodobnější z možných.

Monte Carlo Smoothing for Non-Linearly Distorted Signals

V roce 2001 pak autoři William Fong a Simon Godsill vydali článek *Monte Carlo Smoothing for Non-Linearly Distorted Signals* [10], ve kterém představují metodu založenou na Monte Carlo filtrování a používají ji k vyhlazení nelineárně zkreslených signálů. Metoda je pak aplikována na dekvantizaci signálu a declipping. V případě declippingu, kdy bylo saturováno cca 20 % vzorků, je udávané průměrné zlepšení v SNR 8,1 dB.

Deconvolution and Model-Based Restoration of Clipped Ultrasonic Signals

Statistiku použil i Olofsson v článku [11] z roku 2005, kde aplikoval techniku *maximum a posteriori* na rekonstrukci saturovaných ultrazvukových signálů. V článku představil dvě metody. První vede na problém řešený pomocí kvadratického programování, druhý jednodušší pak k systému lineárních rovnic.

A Simple Algorithm for the Restoration of Clipped Speech Signal

V roce 2007 byl představen algoritmus pro restaurování saturovaných vzorků řečového signálu [12]. Autoři vycházejí z předpokladu, že lidská řeč je vysoce předvídatelná. Jejich algoritmus nejprve vypočítá predikční koeficienty ze známých vzorků signálu a následně pomocí lineární predikce dopočítá chybějící vzorky. Autoři testují algoritmus na lidské i syntetické řeči.

Bohužel však v článku neuvádějí objektivní vyhodnocení výsledků, pouze subjektivní srovnání. Vzorky řečového signálu (jak saturované, tak rekonstruované) byly prezentovány několika posluchačům, kteří hodnotili srozumitelnost řeči na škále od nuly do pěti, kde nula znamená naprosto nesrozumitelné a pětka naopak perfektně srozumitelné. Zatímco hodnocení srozumitelnosti saturovaných signálů bylo v průměru okolo dvou, hodnocení restaurovaných signálů dosahovalo hodnot mezi čtyřkou a pětkou. Z toho lze usoudit, že metoda má pozitivní vliv na kvalitu a srozumitelnost při rekonstrukci saturovaného signálu.

Restoration of Clipped Audio Signal Using Recursive Vector Projection

V roce 2011 byla představena další metoda [13], která rekonstrukci signálu řeší pomocí tzv. rekurzivní projekce vektorů. Algoritmus si nejprve vytvoří matici, jejíž vektory tvoří komplexní sinusové signály s různými frekvencemi. Tyto sinusoidy potom zkouší dosadit do míst, kde je signál saturovaný a počítá se chybový vektor. Autoři zde uvádějí zlepšení hodnoty SNR rekonstruovaného signálu o 2 dB oproti saturovanému, nicméně podle subjektivního srovnání posluchačů nebyl prakticky slyšet rozdíl.

A Constrained Matching Pursuit to Audio Declipping

První algoritmus, který k declippingu audio signálu použil řídkou reprezentaci signálů, byl publikován v článku [14] z roku 2011. Autoři se v něm zaměřili na rekonstrukci obecného audio signálu (nejen např. na řečový signál nebo ultrazvukový signál jako v předchozích článcích).

Celý signál byl zpracováván po blocích, kdy byla použita metoda přičítání přesahu (*OLA*). Bloky byly dlouhé 64 ms (což odpovídá 512 vzorkům pro signály v telefonní kvalitě s vzorkovacím kmitočtem 8 kHz a 1024 vzorkům pro hudební signály s vzorkovacím kmitočtem 16 kHz) a použitý přesah bloků byl 75 %. Jednotlivé bloky potom byly zpracovávány modifikovaným OMP algoritmem (viz část 3.4), který autoři nazvali omezený OMP (*Constrained OMP*, C-OMP). Jako slovník byla použita nadbytečná DCT. Výsledky algoritmu byly porovnány objektivně pomocí SNR na hudebním i řečovém signálu, kde průměrné výsledky dosáhly zlepšení až o 4,5 dB oproti původnímu saturovanému signálu.

Recovering a Clipped Signal in Sparseland

Krátce poté byl vydán další článek [15], který také řeší declipping pomocí řídké reprezentace signálů a porovnává svoje výsledky s předchozím uvedeným. Autoři v něm představují dva algoritmy – první *Reweighted ℓ_1 minimization with clipping constraints* ($R\ell_1CC$) a výpočetně mnohem méně náročný *Trivial Pursuit with Clipping Constraints* (TPCC).

$R\ell_1CC$ funguje na principu ℓ_1 minimalizace, kdy je při každé iteraci řídký vektor násoben váhou, která se v každé iteraci mění podle toho, jak se změnil hledaný vektor oproti předchozí iteraci. Jako slovník je zde použita Diskrétní Fourierova Transformace (DFT).

Algoritmus TPCC také používá DFT a skládá se ze dvou fází. Nejprve vypočítá Fourierovo spektrum signálu a v druhé fázi odhaduje hodnotu jednotlivých koeficientů z původního spektra metodou nejmenších čtverců. Pokud dopředu známe řídkost originálního signálu k , můžeme signál rekonstruovat nalezením k největších harmonických složek saturovaného signálu. V obecném případě, kdy k neznáme, pak algoritmus iterativně rekonstruuje jednotlivé harmonické „hladovým“ způsobem, dokud chyba rekonstrukce na spolehlivých vzorcích není dostatečně malá.

Autoři testují výsledky algoritmů pouze na uměle vytvořených řídkých signálech délky $N = 128$ vzorků a jako měřítko zde používají minimální počet „spolehlivých“, tj. nesaturovaných vzorků vzhledem k řídkosti signálu K . Z uvedených výsledků vychází jako nejlepší algoritmus $R\ell_1CC$, následován TPCC (avšak za cenu výrazně nižší výpočetní náročnosti). Oba algoritmy však v daném kritériu překonaly Constrained OMP představený v článku [14].

Least Squares with Examples in Signal Processing

Webový článek Ivana Salecnicka *Least Squares with Examples in Signal Processing* [16] z roku 2013 se nezaměřuje přímo na audio declipping, ale prezentuje metodu nejmenších čtverců a její následné využití ve zpracování signálů v úlohách lineární

predikce, vyhlazování, dekonvoluce, systémové identifikace a odhadu chybějících dat. Součástí poslední jmenované části je zmíněna také úloha audio-declipping, která je řešena jako interpolace chybějících vzorků v signálu, kdy je minimalizována třetí diference signálu pomocí metody nejmenších čtverců. Výsledek této metody je možné vyjádřit explicitně, proto je výpočet mnohem rychlejší, než konkurenční iterativní metody. Bohužel však nejsou prezentovány žádné výsledky metody.

Consistent Iterative Hard Thresholding for Signal Declipping

Dalším článkem, který se také zaměřuje na declipping pomocí řídkých reprezentací je [17]. Autoři v něm použili optimalizační algoritmus nazvaný *Iterative Hard Thresholding* a rozšířili ho o omezující podmínky pro účely declippingu (*Iterative Hard Thresholding – Declipping Constraints*, IHT-DC). Tento algoritmus je v článku přímo srovnáván s dříve představeným C-OMP, jehož nevýhodou je fakt, že ve své první fázi nebere v úvahu informaci ze saturovaných vzorků signálu. Díky tomu, že tento algoritmus při každé iteraci počítá i s omezujícími podmínkami, podává lepší výsledky, než C-OMP.

Algoritmus byl testován prakticky shodně jako C-OMP, tedy opět byl použit jako slovník DCT, signál byl zpracováván po blocích o délce $N = 1024$ vzorků, což při vzorkovací frekvenci signálu 16 kHz znamená délku bloku 64 ms a přesah bloků byl 75%.

Výsledky algoritmu byly testovány na dvou zvukových nahrávkách pomocí SNR i subjektivním poslechovým hodnocením a porovnávány se softwarem Adobe De-Clipper, s metodou na bázi AR a algoritmem C-OMP. Z uvedených výsledků vyplývá, že IHT-DC podává z testovaných metod nejlepší výsledky, avšak je nutno přihlídnout k faktu, že testování proběhlo pouze na dvou nahrávkách.

Declipping of Audio Signals Using Perceptual Compressed Sensing

Další poměrně zajímavá a inovativní metoda byla představena v roce 2013 v článku [6], která podle názvu slibuje přístup pomocí komprimovaného snímání (viz [18]), ve skutečnosti zde však žádné komprimované snímání neprobíhá. Při řešení minimalizační úlohy zde vystupuje měřicí matice Φ , která zastupuje výběr nesaturovaných vzorků signálu a Ψ je slovník (v tomto případě je použita DFT). Signál je opět zpracováván po blocích o délce $N = 512$ vzorků, avšak je použito Hannovo okno místo obdélníkového okna jako v předchozích metodách.

Zajímavým vylepšením metody je hlavně implementace psychoakustického modelu vnímání zvuku. Algoritmus si nejprve vypočítá okamžitý maskovací práh aktuálního zpracovaného bloku podle standardu MPEG-1 Layer 1, který potom použije pro následující blok signálu v minimalizační úloze, kde místo hledaného řídkého

vektoru vystupuje součin $\mathbf{P}\mathbf{z}$. Matice \mathbf{P} obsahuje na hlavní diagonále reciproké hodnoty okamžitého maskovacího prahu, které tedy označují relativní vjemovou důležitost jednotlivých komponent signálu. Použití váhové matice \mathbf{P} upřednostňuje použití komponent, které mají vyšší maskovací práh. Tento přístup je žádoucí, protože potenciálně výrazně slyšitelné signálové složky, které nejsou přítomny v původním signálu, ale mohly by se objevit v rekonstruovaném signálu jsou potlačeny a naopak méně slyšitelné složky signálu jsou tolerovány ve větší míře. Navíc je také podpořena rekonstrukce signálových složek, které jsou vjemově důležité.

Autoři v článku vytvořili celkem tři algoritmy – algoritmus CSL0 využívá v minimalizační úloze ℓ_0 -normu, CSL1, který využívá ℓ_1 -normu a konečně PCSL1, který také využívá ℓ_1 -normu a navíc přidává psychoakustický model. Jejich porovnání potom proběhlo na několika zvukových ukázkách porovnáním ukazatele SNR, dále pomocí ukazatele PEAQ ODG [19] (tento ukazatel by měl lépe vyhodnotit kvalitu zvukového signálu korespondující s lidským slyšením) a samozřejmě také subjektivními testy.

U všech třech testovaných algoritmů došlo po rekonstrukci daným algoritmem ke zvýšení hodnot SNR i PEAQ ODG. V porovnání SNR si nejlépe vedl CSL0, avšak podle ukazatele PEAQ ODG a i podle subjektivních testů dominoval PCSL1. Zde je dobře vidět, že SNR ne vždy dobře koresponduje s lidským vnímáním zvukového signálu.

Audio Declipping with Social Sparsity

V roce 2014 vyšel další článek řešící declipping pomocí řídké reprezentace [20], který je významný především zakomponováním tzv. sociální řídkosti do úlohy declippingu. Autoři v něm pro omezení množiny přípustných řešení použili *hinge squared* funkci a pro vyřešení optimalizačního problému používají relaxovaný (F)ISTA algoritmus.

Vylepšením předchozích algoritmů je i použití framu namísto báze jako slovníku. V tomto případě byl použit těsný Gaborův frame založený na Hannově posuvném okně o délce 1024 vzorků (64 ms při vzorkovací frekvenci 16 kHz).

Článek se také zaměřuje na porovnání několika *shrinkage* operátorů – *Lasso* (L), *Windowed Group-Lasso* (WGL), *Empirical Weiner* (EW) a *Persistent Empirical Weiner* (PEW). Výsledky jsou testovány na hudebním i řečovém signálu pro několik úrovní clippingu a porovnáváno bylo zlepšení SNR oproti saturovanému signálu. Nejlepší výsledky z testovaných operátorů podává PEW, který dosáhl průměrného zlepšení při úrovni clippingu $\theta_c = 0,6$ o 14 dB v případě řečového signálu a o 10 dB v případě hudebního signálu.

Zvukové ukázky výsledků tohoto algoritmu jsou k dispozici na URL [21].

Audio Declipping by Cosparse Hard Thresholding

Krátce po předchozím představeném článku byl zveřejněn další [22], který vychází z dříve představeného algoritmu *Consistent Iterative Hard Thresholding*, ale namísto syntezujícího modelu je zde použit analyzující model. Autoři museli především vyřešit problém s projekcí, která se v případě použití analyzujícího modelu stává NP-těžkou. Problém však obešli zvolením jiného přístupu a použili metodu založenou na *Alternating Direction Method of Multipliers*. Svůj algoritmus pak nazvali CoDecHT (*COsparse DEClipping by Hard Thresholding*).

Výsledky metody byly testovány velmi podobně jako u předchozích metod, tzn. bylo měřeno zlepšení SNR oproti saturevanému signálu, avšak pouze na dvou 10sekundových hudebních vzorcích. Zpracování proběhlo opět po blocích metodou OLA s parametry stejnými jako v případě Consistent IHT. V článku byl tento algoritmus porovnáván s původním algoritmem Consistent IHT a také s předchozím představeným algoritmem *Social Sparsity*. Výsledky algoritmů *Social Sparsity* a CoDecHT byly podobné, avšak lepší, než v případě původního algoritmu Constrained-IHT.

Declipping of Speech Signals Using Frequency Selective Extrapolation

V tomto článku [23] byla pro účel declippingu použita metoda *Frequency Selective Extrapolation* (FSE), která se zatím běžně používá pro odstraňování chyb, popř. rekonstrukci nekompletních informací v obraze. FSE iterativně generuje model signálu, který je tvořen superpozicí Fourierových bázeových funkcí. Saturevané vzorky signálu pak mohou být nahrazeny odhadovanými vzorky z modelu.

Výsledky metody pak byly testovány na řečových signálech o délce pět sekund s vzorkovacím kmitočtem 16 kHz. Porovnání zde probíhalo podle objektivního ukazatele SNR a výsledky byly srovnány s metodou OMP používající Gaborův slovník (označení OMP-G) a jednoduchou metodou založenou na autoregresi.

Prezentované výsledky představené metody uvádějí maximální zlepšení v SNR až o 3,52 dB a průměrné zlepšení o 1,01 dB. Z porovnávaných metod tedy FSE podává nejlepší výsledky. Zajímavé je i porovnání z hlediska časové náročnosti. V případě úrovně omezení $\theta_c = 0,5$ je doba zpracování FSE 13,4 s. Naopak u metody OMP-G je to cca 495 s.

Least Squares Signal Declipping for Robust Speech Recognition

Další článek řešící declipping audiosignálu je [24], který se zaměřuje na rekonstrukci řečového signálu a to z důvodu zlepšení přesnosti automatického rozpoznání řeči. Autoři zde uvádějí, že závislost mezi počtem saturevaných vzorků v signálu a chybou rozpoznání slova je prakticky lineární. Tato metoda se inspirovala některými předchozími autoregresivními metodami, které však mají tu nevýhodu, že provádějí

pouze interpolaci vzorků signálu a tudíž nedodržují omezující podmínky. Použitím přístupu metody nejmenších čtverců a explicitním dodáním omezujících podmínek pak vzniká algoritmus CBAR (*Constrained Blind Amplitude Reconstruction*).

Představený algoritmus provádí minimalizaci ℓ_2 -normy druhé derivace vstupního signálu (čímž se vynutí hladkost průběhu signálu), avšak vzhledem k omezujícím podmínkám. Pro vyřešení tohoto optimalizačního problému je v článku použita kvazi-Newtonova metoda. CBAR zpracovává signál po blocích o délce 80 vzorků bez překryvu, avšak pokud se konec zpracovávaného bloku vyskytuje v oblasti, kde je signál satureován, délka bloku se zvětší o potřebný počet vzorků. Autoři dodávají, že pro tuto aplikaci lze použít i jinou než druhou derivaci, nicméně v praxi se druhá derivace osvědčila nejlépe.

Výsledky této metody nejsou prezentovány pomocí SNR, ale uvádí se zde chybovost v automatickém rozpoznání řeči v závislosti na počtu satureovaných vzorků (*Word Error Rate* – WER). Pro každou testovanou úroveň zkreslení vstupního signálu je zde pak vyhodnoceno WER pro satureovaný signál, výsledek metody CBAR a výsledek metody C-IHT. V uvedeném porovnání podává metoda CBAR poměrně dobré výsledky. Při 25 % satureovaných vzorků vstupního signálu dochází vlivem metody CBAR o zlepšení WER až o 56 %, což je o 24 % lepší výsledek, než v případě C-IHT.

Efficient Audio Declipping Using Regularized Least Squares

Krátce po předchozím představeném článku, ve kterém byla popsána metoda CBAR, autoři vydali další článek [25], v němž se zaměřují na vylepšení metody CBAR především z pohledu výpočetní rychlosti. Jejich snahou je při zachování kvality rekonstrukce metody CBAR dosáhnout takové výpočetní rychlosti, aby bylo možné real-time zpracování na běžně výkonném laptopu. Podobně jako v jejich předchozím článku, i zde se zaměřují především na declipping zvukových signálů z důvodu zlepšení přesnosti automatického rozpoznání řeči. Metoda představená v tomto článku byla nazvána RBAR (*Regularized Blind Amplitude Reconstruction*).

Autoři zde uvádějí, že nízká rychlost zpracování v případě metody CBAR byla dána tvrdou omezující podmínkou. V případě metody RBAR pak byla úloha převedena na neomezený tvar a omezující podmínka má pak v minimalizační úloze podobu dvou regularizačních členů. Tento krok vedl k rapidnímu zrychlení metody. Pro srovnání autoři uvádějí, že zatímco metoda CBAR v některých případech zabrala až 400násobek délky trvání zpracovávaného signálu, metoda RBAR je v průměru pětikrát rychlejší, než doba trvání signálu. RBAR lze tedy bez větších problémů použít pro zpracování v reálném čase.

Metoda RBAR zpracovává signál po blocích o délce 50 ms každých 12,5 ms a k re-

konstrukci zpracovaných bloků je použita metoda přičítání přesahu (OLA). Protože představená metoda má tendenci vyhlazovat frikativa (neboli třené souhlásky), před samotným výpočtem metoda provádí keprální analýzu, pomocí níž detekuje pouze znělé bloky, které následně zpracovává. Bloky obsahující frikativa tedy zůstanou nepozměněny a declipping se provádí pouze na blocích obsahující znělé hlásky.

Výsledky metody pak byly vyhodnocovány opět pomocí WER při automatickém rozpoznání hlasu, ale také klasicky pomocí SNR a srovnány s metodami C-IHT a CBAR. I když výstupy metody RBAR nepodávají tak dobré výsledky jako CBAR, popř. C-IHT (jak při WER, tak i SNR), v obou srovnáních metoda RBAR přináší zlepšení oproti saturovanému signálu. Hlavní výhodou této metody je však možnost zpracování v reálném čase.

Audio Declipping via Nonnegative Matrix Factorization

V roce 2015 byl publikován další článek [26], který řeší současně problematiku doplňování chybějících vzorků signálu (*audio inpainting*), tak i problematiku declippingu. Představená metoda zde využívá princip faktorizace nezápornými maticemi, neboli *non-negative matrix factorization* (NMF). Princip NMF byl pro účely inpaintingu, resp. declippingu v tomto článku použit vůbec poprvé, nicméně již dříve bylo NMF ve zpracování signálů využito např. pro separaci zdrojů.

Autoři zde použili kombinaci algoritmu GEM (*generalized expectation maximization*) pro odhad parametrů modelu a Wienerova filtrování pro odhad rekonstrukce signálu podle parametrů modelu. Tento iterativní algoritmus nejprve náhodně nastaví nezáporné matice \mathbf{W} a \mathbf{H} , které následně použije ve Wienerově filtraci pro odhad původního signálu. V dalším kroku je pak tento odhad použit pro aplikování omezujících podmínek pro declipping. V posledním kroku algoritmu pak dochází k aktualizaci matic \mathbf{W} a \mathbf{H} .

Autoři v článku uvádějí čtyři možné přístupy pro aplikování omezujících podmínek. První přístup je nazván neomezený (*unconstrained*) a kompletně ignoruje úroveň clippingu. Declipping se tedy v tomto případě mění na inpainting. Druhý přístup v průběhu iterací také neřeší omezující podmínky a vynutí je až na konci (v post-processingu). Třetí přístup, nazvaný signálová projekce, pak upravuje odhadovaný signál při každé iteraci tak, aby odpovídal omezujícím podmínkám. Nicméně u tohoto případu není takto upravena kovarianční matice, se kterou algoritmus také pracuje. Posledním případem je pak kovarianční projekce, která v průběhu iterací vhodně aktualizuje i kovarianční matici. V tomto případě pak ale dochází k odhadu vzorků celého signálu, ne pouze na saturovaných vzorcích jako v předchozích případech.

Výsledky metody byly testovány na deseti hudebních a deseti řečových signálech

podle hodnoty SNR, která byla počítána pouze na saturovaných vzorcích. V algoritmu použitá STFT byla počítána přes sinová okna s polovičním překryvem o délce 1024 vzorků (64 ms pro vzorkovací kmitočet 16 kHz) a představený algoritmus provedl 50 iterací. Pro testování algoritmu byly použity všechny čtyři přístupy aplikace omezujících podmínek popsané výše a byly porovnány s metodami IHT, C-OMP a Social Sparsity.

Z prezentovaných výsledků je patrné, že z představených přístupů podává nejlepší rekonstrukci verze algoritmu s kovarianční projekcí, která má velmi podobné výsledky jako algoritmus Social Sparsity. Všechny přístupy NMF metody však překonávají metody C-OMP a IHT.

Sparsity and Cosparsity for Audio Declipping: A Flexible Non-convex Approach

Nejnovější článek na téma declipping zvukového signálu je [27] z roku 2015. Jeden z autorů (Srđan Kitić) pracoval už na dříve představených algoritmech C-IHT a CoDecHT.

Článek představuje algoritmus založený na nekonvexní heuristice, která může být parametrizována pro použití jak na syntezujícím, tak analyzujícím modelu signálu. Autoři své algoritmy nazvali souhrnně SPADE (*S*Parse *A*udio *D*Eclipper a rozlišují, zda se pracuje se syntezujícím modelem signálu (S-SPADE) nebo s analyzujícím modelem (A-SPADE).

Po představení vlastního algoritmu v pseudokódu se článek zabývá také volbou slovníků z hlediska výpočetní náročnosti. Například pro verzi A-SPADE je výhodné, pokud slovník \mathbf{A} tvoří těsný frame. Dále je vhodné poznamenat, že oba algoritmy jsou v případě volby navzájem inverzních slovníků (tj. když \mathbf{A} je báze) identické.

Porovnání výsledků těchto algoritmů pak probíhá s dříve představenými C-IHT a *Audio Declipping with Social Sparsity*. Pro všechny experimenty byl jako slovník zvolen STFT, Hammingovo okno s délkou 1024 vzorků a 75% překrytí. Porovnání výsledků proběhlo pomocí ukazatele SNR, avšak je počítán rozdíl mezi saturovaným a rekonstruovaným signálem a pouze na saturovaných vzorcích signálu.

Výsledky byly testovány na 10 hudebních vzorcích se vzorkovací frekvencí 16 kHz a 16bitovým kódováním. Testování proběhlo nejen v závislosti na úrovni zkreslení vstupního signálu, ale také na míře nadbytečnosti slovníku. Nejlépe ze všech algoritmů dopadl S-SPADE, následován A-SPADE, Social Sparsity a C-IHT. Nicméně autoři také podotýkají, že i když S-SPADE podává nejlepší výsledky, jeho konvergence je v řádu hodin. Naopak A-SPADE, jehož výsledky jsou pouze mírně horší, nalezne uspokojivý výsledek (v průměru) za tři minuty a je tak nejrychlejším algoritmem ze všech testovaných.

Výsledky algoritmů si lze poslechnout na webové stránce [28].

3 ÚVOD DO ŘÍDKÉ REPREZENTACE

Jak bylo vidět v předchozí části práce, metody založené na řídkých reprezentacích podávají velmi dobré výsledky, nicméně stále je zde určitý prostor ke zlepšení. Tato kapitola bude věnována základnímu úvodu do řídké reprezentace a budou zde objasněny pojmy, se kterými bude pracováno v navazující kapitole, která se už bude přímo věnovat declippingu.

3.1 Základní pojmy a značení

V této sekci budou definovány základní pojmy a značení, které čtenáři usnadní orientaci a pochopení dalších částí práce.

Skalární veličiny budou značeny kurzívou, např. m , N . Naopak vektory budou tučným řezem malými písmeny, např. \mathbf{x} , \mathbf{y} , matice pak tučně velkými písmeny, tedy např. \mathbf{A} , \mathbf{B} . Pokud nebude uvedeno jinak, vektory budeme uvažovat jako sloupcové s indexací prvků počínaje jedničkou. Vektor \mathbf{x} tak můžeme zapsat jako $\mathbf{x} = [x_1, x_2, \dots, x_n]^\top$. Počet prvků množiny (tzv. kardinalitu množiny) pak budeme značit obdobně jako absolutní hodnotu, tj. např. $|\{-5, 2, 7, 1, -8\}| = 5$.

Dále definujeme pojem nosič vektoru. Tímto označením máme na mysli množinu indexů, v nichž má daný vektor nenulové hodnoty. Tuto množinu pak značíme $\text{supp}(\mathbf{x})$ – z anglického *support*. Obecně tedy platí $\text{supp}(\mathbf{x}) = \{i \mid x_i \neq 0\}$ [29].

Mějme tedy řídký signál $\mathbf{x} = [0, 0, 3, 0, 0, 4, 5, 0, 0, 8, 0, 1]^\top$. Podle výše uvedených definic bude platit $\text{supp}(\mathbf{x}) = \{3, 6, 7, 10, 12\}$ a $|\text{supp}(\mathbf{x})| = 5$.

Pro pochopení řídkých reprezentací je tak nutné definovat ℓ_p -normu vektoru, která je pro libovolný vektor $\mathbf{x} \in \mathbb{C}^N$ definována jako

$$\begin{aligned} \|\mathbf{x}\|_p &:= \left(\sum_{i=1}^N \|x_i\|^p \right)^{1/p} \quad \text{pro } 1 \leq p < \infty, \\ \|\mathbf{x}\|_p &:= \sum_{i=1}^N |x_i|^p \quad \text{pro } 0 < p < 1, \\ \|\mathbf{x}\|_\infty &:= \max_i |x_i|, \\ \|\mathbf{x}\|_0 &:= |\text{supp}(\mathbf{x})|. \end{aligned} \tag{3.1}$$

Přísně vzato se o ℓ_p -normu vektoru jedná pouze v případě $1 \leq p < \infty$. Pro zjednodušení však budeme používat označení ℓ_p -norma pro všechna p [29].

Pravděpodobně nejčastěji používanou ℓ_p -normou je ℓ_2 -norma, což je vlastně klasická Euklidovská vzdálenost vektoru.

Za zmínku však stojí speciální případy ℓ_p -norm, jako jsou tzv. ℓ_1 -norma (nazývaná také jako *Manhattan*), tedy $\|\cdot\|_1$, která představuje součet absolutních hodnot

prvků vektoru a norma $\|\cdot\|_0$, reprezentující počet nenulových složek vektoru. Používaná je také norma $\|\cdot\|_\infty$, která vrací prvek vektoru s největší hodnotou [29].

Při práci s řídkými reprezentacemi je také důležité definovat pojem řídkost. Vektor $\mathbf{x} \in \mathbb{C}^N$ nazveme k -řídkým, pokud platí

$$\|\mathbf{x}\|_0 \leq k. \quad (3.2)$$

Jinými slovy pokud je počet nenulových prvků vektoru menší nebo roven konstantě k , pak můžeme o vektoru \mathbf{x} tvrdit, že je k -řídký. Relativní řídkostí potom rozumíme poměr počtu nenulových složek vektoru k ku celkové délce vektoru N , tedy $\frac{k}{N}$ [29].

3.2 Syntezující a analyzující model signálu

Protože v další části práce budu často pracovat s transformacemi a aditivním modelem signálu, v této části vysvětlím, co to vlastně znamená. Zmíním zde i analyzující model, který používají například některé pokročilejší metody z části 1.3.

Diskrétní signál můžeme mimo jiné vyjádřit ve formě součinu matice a vektoru souřadnic (za předpokladu, že matice je v tomto případě báze, popř. frame). Proces převedení signálu na tvar tohoto součinu lze také nazvat transformací. Použitá matice se pak obvykle označuje jako transformační matice. Jako příklad lze použít obecně známou Fourierovu transformaci, díky které lze vyjádřit signál pomocí součtu sinusových a kosinusových funkcí s různým kmitočtem. Transformační matice pak obsahuje jednotlivé sinusoidy a kosinusoidy, a vektor souřadnic pak jednotlivé koeficienty transformace.

Mějme libovolný signál \mathbf{y} . Tento signál je možné podle výše uvedeného odstavce vyjádřit jako součin transformační matice (označme si ji \mathbf{A}) a vektoru souřadnic \mathbf{x} , platí tedy, že $\mathbf{y} = \mathbf{A}\mathbf{x}$. Takovýto model označujeme jako aditivní (popř. také syntezující), protože výsledný signál skládáme z více částí. Pokud v tomto tvaru předpokládáme řídký vektor \mathbf{x} , je tento model v anglické literatuře označován jako *sparse synthesis*.

Existuje však i opačný (a méně používaný) pohled na problematiku. Namísto skládání celkového signálu z více vektorů můžeme naopak koeficienty signálu teprve hledat. Jinými slovy analýzou signálu \mathbf{y} vznikne řídký vektor \mathbf{z} . Platí tedy, že $\mathbf{\Omega}\mathbf{y} = \mathbf{z}$ [30]. Tento model pak v případě řídkého \mathbf{z} nazýváme analyzující, v anglické literatuře označován jako *sparse analysis*, popř. *co-sparse analysis*.

Oba modely (syntezující a analyzující) jsou identické, pokud použité matice jsou čtvercové navzájem inverzní matice, tedy $\mathbf{A} = \mathbf{\Omega}^{-1}$ [30]. Jinými slovy matice v \mathbb{R}^N tvoří báze.

Teorie v následujících částech práce je popsána pomocí syntezujícího (neboli aditivního) modelu.

3.3 Hledání řídkého řešení

Při hledání řídkého řešení prakticky řešíme obvyklou soustavu lineárních rovnic $\mathbf{Ax} = \mathbf{y}$, ale s tím rozdílem, že chceme, aby vektor \mathbf{x} byl co nejřidší. Tuto úlohu lze vyjádřit následovně:

$$\min_{\mathbf{x}} \|\mathbf{x}\|_0 \quad \text{vzhledem k} \quad \mathbf{Ax} = \mathbf{y}, \quad (3.3)$$

kde známe vektor $\mathbf{y} \in \mathbb{C}^m$ a matici $\mathbf{A} \in \mathbb{C}^{m \times N}$. Budeme uvažovat pouze případy, kdy $m < N$ popř. $m \ll N$ a \mathbf{A} je plně řádkové hodnosti. Matici \mathbf{A} obvykle označujeme jako slovník (*dictionary*) a její sloupce jako atomy (*atoms*) [29].

V praxi se často může stát, že zpracovávaný signál bude zašuměn a právě tento šum způsobí odchylku od přesného řešení. V takovém případě při výpočtu $\mathbf{Ax} = \mathbf{y}$ povolíme malou odchylku. Tuto úlohu pak zapíšeme

$$\min_{\mathbf{x}} \|\mathbf{x}\|_0 \quad \text{vzhledem k} \quad \|\mathbf{Ax} - \mathbf{y}\|_p \leq \delta, \quad (3.4)$$

kde většinou uvažujeme $p = 2$ [29].

Nalezení řídkého řešení není jednoznačná záležitost, v některých případech dokonce řídké řešení neexistuje. Existují však postačující podmínky, které jsou schopny určit, zda řídké řešení existuje, popř. jestli je toto řešení jedinečné a nejřidší možné.

Důležitou vlastností matice pro rozhodování o existenci řídkého řešení a jeho vlastnostech je číslo $\text{spark}(\mathbf{A})$, které udává nejmenší počet sloupců matice \mathbf{A} , jenž jsou lineárně závislé. Formálně můžeme zapsat jako [29]:

$$\text{spark}(\mathbf{A}) = \min_{\mathbf{z} \in \ker \mathbf{A}, \mathbf{z} \neq 0} \|\mathbf{z}\|_0. \quad (3.5)$$

Z definice vyplývá, že pro nenulovou matici $\mathbf{A} \in \mathbb{C}^{m \times N}$, kde $m < N$, platí, že číslo spark může nabývat hodnot $\text{spark}(\mathbf{A}) \in \{2, \dots, m+1\}$. Pro řešení soustavy $\mathbf{Ax} = \mathbf{y}$ pak platí, že pokud řešení \mathbf{x} splňuje podmínku

$$\|\mathbf{x}\|_0 < \frac{\text{spark}(\mathbf{A})}{2}, \quad (3.6)$$

pak je toto řešení jedinečné a zároveň nejřidší možné [29].

Číslo $\text{spark}(\mathbf{A})$ se však v praxi příliš nepoužívá, a to z toho důvodu, že jeho nalezení je výpočetně stejně náročné, jako samotné řešení problému (3.3).

Pro naše účely můžeme použít i jinou vlastnost matice \mathbf{A} a tou je tzv. vzájemná koherence (*mutual coherence*). Je definována jako největší absolutní normalizovaný skalární součin dvou různých sloupců matice \mathbf{A} ,

$$\mu(\mathbf{A}) = \max_{1 \leq j, k \leq N, j \neq k} \frac{|\mathbf{a}_j^\top \mathbf{a}_k|}{\|\mathbf{a}_j\|_2 \cdot \|\mathbf{a}_k\|_2}, \quad (3.7)$$

kde \mathbf{a}_j označuje j -tý sloupec matice \mathbf{A} . Pro libovolnou matici \mathbf{A} platí mezi číslem spark a vzájemnou koherencí matice následující vztah:

$$\text{spark}(\mathbf{A}) \leq 1 + \frac{1}{\mu(\mathbf{A})}. \quad (3.8)$$

Z výše uvedeného vyplývá, že pokud pro soustavu $\mathbf{Ax} = \mathbf{y}$ najdeme řešení \mathbf{x} splňující

$$\|\mathbf{x}\|_0 < \frac{1}{2} \left(1 + \frac{1}{\mu(\mathbf{A})} \right), \quad (3.9)$$

pak je \mathbf{x} nutně nejřidší možné a zároveň jedinečné. Dokonce platí, že takového řešení můžeme nalézt ℓ_1 -minimalizací (viz níže). Pro nalezení co nejřidšího řešení pak plyne potřeba použít maximálně nekoherentní slovníky [29].

Nyní předpokládejme, že existuje k_0 -řádké řešení soustavy, o kterém víme, že je jednoznačné a nejřidší možné. Pokud bychom trvali na nalezení přesného řešení podle (3.3), museli bychom projít všech $\binom{N}{k_0}$ možností. To odpovídá složitosti NP-hard, což v praxi, kdy N nabývá vysokých hodnot, není přijatelné [29]. Navíc ℓ_0 norma není konvexní a proto ani nemůžeme použít žádnou z metod konvexní optimalizace. Nabízí se však možnost nekonvexní ℓ_0 normu nahradit nejbližší normou, která už konvexní je – ℓ_1 -normou. Tuto aproximační úlohu můžeme zapsat jako

$$\min_{\mathbf{x}} \|\mathbf{x}\|_1 \quad \text{vzhledem k} \quad \mathbf{Ax} = \mathbf{y}. \quad (3.10)$$

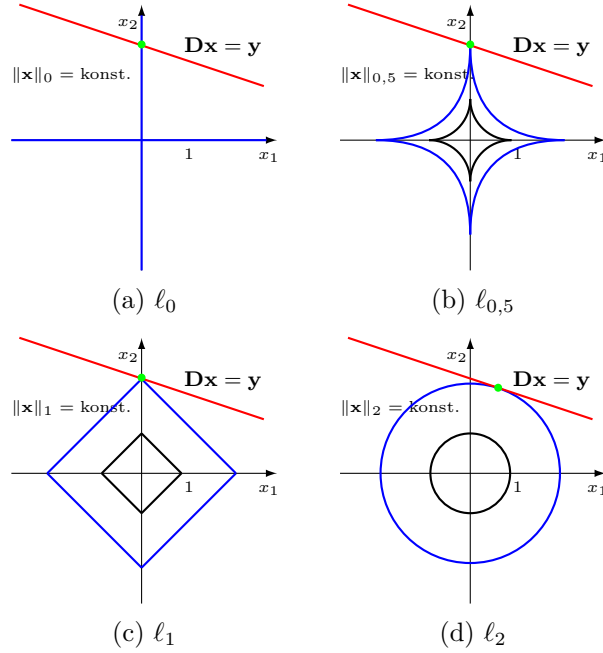
Cenou za rychlejší výpočet je tak mírná nepřesnost řešení [29].

Nalezení řešení úlohy (3.10) nezaručuje, že nalezené řešení bude nejřidší. V praxi se však ukazuje, že ve většině případů se řešení ℓ_1 -relaxace shoduje s řešením ℓ_0 minimalizace. Pouze ve výjimečných případech může nastat situace, kdy pomocí ℓ_1 -minimalizace nalezneme prakticky nekonečně mnoho řešení, avšak ta „nejřidší“ budou pouze ta krajní.

Pro ilustraci ekvivalence ℓ_0 - a ℓ_1 minimalizace slouží obrázek 3.1, který zobrazuje jednotkové koule v normách (a) ℓ_0 , (b) $\ell_{0,5}$, (c) ℓ_1 a (d) ℓ_2 a jejich dotyk s nadrovinou určenou soustavou $\mathbf{Ax} = \mathbf{y}$. V tomto případě je jasně patrné, že ℓ_0 - i ℓ_1 -minimalizace naleznou stejné řešení, na rozdíl od normy ℓ_2 , která nalezne řešení s nejmenší energií – toto řešení však není řádké. Pokud by nadrovina procházela právě jednou stranou jednotkové koule pro ℓ_1 -normu, pak by řešení ℓ_1 -relaxace bylo nekonečně mnoho, nejřidší by však byla pouze dvě krajní řešení (tedy průsečíky s osami x_1 a x_2).

Zajištění ekvivalence řešení ℓ_0 - a ℓ_1 -minimalizace je stále předmětem zkoumání. Existuje několik podmínek, jejichž splnění tuto ekvivalenci zajistí, v praxi se však příliš nepoužívají. Pro zajímavost zde zmíním alespoň dvě zajímavé podmínky.

První podmínkou je tzv. vlastnost nulového prostoru (*NSP – Null Space Property*) [31]. Pokud libovolný vektor z jádra $\ker(\mathbf{A})$ splní podmínku NSP, pak existuje k -řádké řešení a zároveň ho lze nalézt ℓ_1 -minimalizací. Druhou podmínkou je pak



Obr. 3.1: Ilustrace jednotkových koulí.

tzv. vlastnost zeslabené izometrie (*RIP – Restricted Isometry Property*) [32], která je oproti NSP výpočetně jednodušší, navíc je stabilní i pod vlivem šumu [29].

3.4 Algoritmy pro hledání řídkých řešení

V této části budou pouze shrnuty základní algoritmy pro nalezení řídkého řešení. Primárním cílem práce však není srovnávat vlastnosti jednotlivých algoritmů, proto zde jednotlivým algoritmům nebude věnována taková pozornost.

Obecně se dají všechny známé a použitelné algoritmy pro hledání řídkého řešení rozdělit do dvou hlavních skupin. První skupina jsou algoritmy, které vycházejí z ℓ_1 -relaxace (viz výše), proto také tuto skupinu algoritmů můžeme označovat jako relaxační. Z jednotlivých zástupců můžeme jmenovat například BP (*Basis Pursuit*), modifikovaný LARS (*Least Angle Regression, homotopy method*), IRLS (*Iterative Reweight Least Squares*) nebo *Dantzig Selector*.

Druhou skupinu pak tvoří tzv. hladové (*greedy*) algoritmy. Jejich hlavní princip spočívá v nalezení alespoň jednoho „nejvýznamnějšího“ atomu v každé iteraci. V každém dalším průběhu už vybraný atom nemůže být zbaven podílu na konečném řešení. Mezi výhody těchto algoritmů patří obecně nízká složitost. Nevýhodou je však fakt, že není zaručeno nalezení globálního optima. Jako hlavního zástupce této skupiny lze jmenovat algoritmus MP (*Matching Pursuit*), popř. v současnosti velmi používaný OMP (*Orthogonal Matching Pursuit*).

Mimo tyto dvě skupiny pak existují i algoritmy založené na prahování (*thresholding*), popř. algoritmy vhodně kombinující vlastnosti hladových i relaxačních algoritmů. Tyto algoritmy jsou pak nazývány hybridní [29].

4 BÁZE A FRAMY

Při zpracování zvukových signálů takovéto signály většinou chápeme jako vektory o konečné délce N , což s výhodou umožňuje využití poznatků z lineární algebry. Moderní metody zpracování signálů pak pracují s reprezentacemi signálů pomocí nedourčených systémů lineárních rovnic (viz část 3.3). Jako reprezentační systém je v těchto případech použita většinou báze vektorového prostoru, popř. redundantní reprezentační systém neboli frame.

Tato kapitola je tedy věnována popisu reprezentačních systémů. Budou zde vysvětleny pojmy jako vektorový prostor, báze, frame, atd. V závěru kapitoly jsou pak popsány dva typy framů (resp. transformací), které jsou v této diplomové práci využity pro úlohu declippingu – Gaborův frame a Constant-Q.

4.1 Vektorový prostor a generátory VP

Nejdříve se zaměříme na vektorový prostor (VP), který si definujeme jako algebraickou strukturu, která splňuje tzv. axiomy vektorového prostoru. Základními prvky vektorového prostoru jsou vektory, nad kterými jsou definovány operace s vektory, jako sčítání, násobení, atd... V tomto případě budeme uvažovat vektory konečné délky N , a proto také dimenze vektorových prostorů bude konečná.

Systémem generátorů vektorového prostoru \mathbb{V} pak myslíme podmnožinu vektorů \mathbf{E} ve \mathbb{V} , která jej generuje. Jinými slovy každý vektor $\mathbf{x} \in \mathbb{V}$ je lineární kombinací generujících vektorů. Pokud je generátorů \mathbb{V} více, než je dimenze prostoru, pak vektor $\mathbf{x} \in \mathbb{V}$ lze zapsat jako vzájemně různé lineární kombinace. Jeden vektor pak může mít v prostoru více reprezentací. Takovýto systém se nazývá *nedourčený*. Pokud \mathbf{x} lze v systému jeho generátorů $\mathbf{E} = \{\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_m\}$ zapsat jako

$$\mathbf{x} = c_1\mathbf{e}_1 + c_2\mathbf{e}_2 + \dots + c_m\mathbf{e}_m = \mathbf{E}\mathbf{c}, \quad (4.1)$$

pak skaláry c_i označujeme jako souřadnice \mathbf{x} v \mathbf{E} [33]. Toto vyjádření koresponduje se syntetizujícím modelem signálu popsaným v části 3.2.

4.2 Báze vektorového prostoru

Bází vektorového prostoru se rozumí minimální systém jeho generátorů. Je to množina vektorů z vektorového prostoru, jejichž lineární kombinací jsme schopni vyjádřit jakýkoli vektor z VP. Báze vektory jsou nutně lineárně nezávislé [33].

Velmi často se v praxi využívají báze *ortogonální* a *ortonormální*, především z důvodu relativní snadnosti práce s takovými bázemi. Ortogonální báze je taková

báze, jejíž všechny dvojice báзовých vektorů jsou na sebe kolmé. Pro libovolné dva vektory z báze $\mathbf{B} = \{\mathbf{b}_1, \dots, \mathbf{b}_n\}$ platí

$$\langle \mathbf{b}_i, \mathbf{b}_j \rangle = 0, \quad \langle \mathbf{b}_i, \mathbf{b}_i \rangle \neq 0. \quad (4.2)$$

Lomená závorka v tomto případě značí skalární součin. V ortonormální bázi navíc pro všechny její prvky platí $\|\mathbf{b}_i\| = 1$, neboli $\mathbf{B}^* = \mathbf{B}^{-1}$ [33].

4.3 Framy

Bázi v konečněrozměrném prostoru \mathbb{V} tvoří lineárně nezávislé vektory, jejichž počet je roven dimenzi prostoru. Pokud však k bázi přidáme další vektory (počet generátorů tedy bude vyšší, než dimenze prostoru), stále jsme schopni reprezentovat jakýkoli vektor ve VP, avšak generátory již budou nutně lineárně závislé. Uvedenou množinu pak nazýváme *frame* [33].

Přidaná nadbytečnost ve framech není nutně na škodu. S výhodou se používají pro svou flexibilitu, protože jsou méně omezené než báze. Nevýhodou jsou však náročnější výpočtové metody a riziko numerické nestability [33].

Aby nadbytečný systém generátorů ve vektorovém prostoru tvořil frame, musí pro konstanty $0 < A \leq B < \infty$ platit

$$A\|\mathbf{x}\|^2 \leq \sum_{k \in \mathbb{I}} |\langle \mathbf{x}, \mathbf{F}_k \rangle|^2 \leq B\|\mathbf{x}\|^2, \quad \forall \mathbf{x} \in \mathbb{V}. \quad (4.3)$$

Prvky framu \mathbf{F}_k se obvykle nazývají *atomy*. Konstanty A, B se nazývají mezemi framu. Framy, pro které platí $A = B$, jsou označovány jako *těsné*. Tutéž vlastnost mají i ortogonální báze, což je v některých případech velmi výhodné. Pokud navíc platí, že $A = B = 1$, nazýváme takové framy jako *Parsevalovy těsné framy*. Dále pro těsné framy platí, že frame operátor $\mathbf{F}\mathbf{F}^* = A \cdot \mathbf{Id}$. Frame operátor těsných framů je tedy diagonální, pro Parsevalovy těsné framy je to dokonce identita.

4.4 Gaborova transformace a framy

Nevýhodou klasické Fourierovy transformace je její globální dosah – jeden koeficient ovlivní podobu celého signálu. To však neodpovídá lidskému vnímání zvukového signálu, kdy jsme schopni vnímat kmitočtovou strukturu signálu, ale zároveň ji lokalizovat v čase. Odtud vznikla potřeba krátkodobé spektrální analýzy [33].

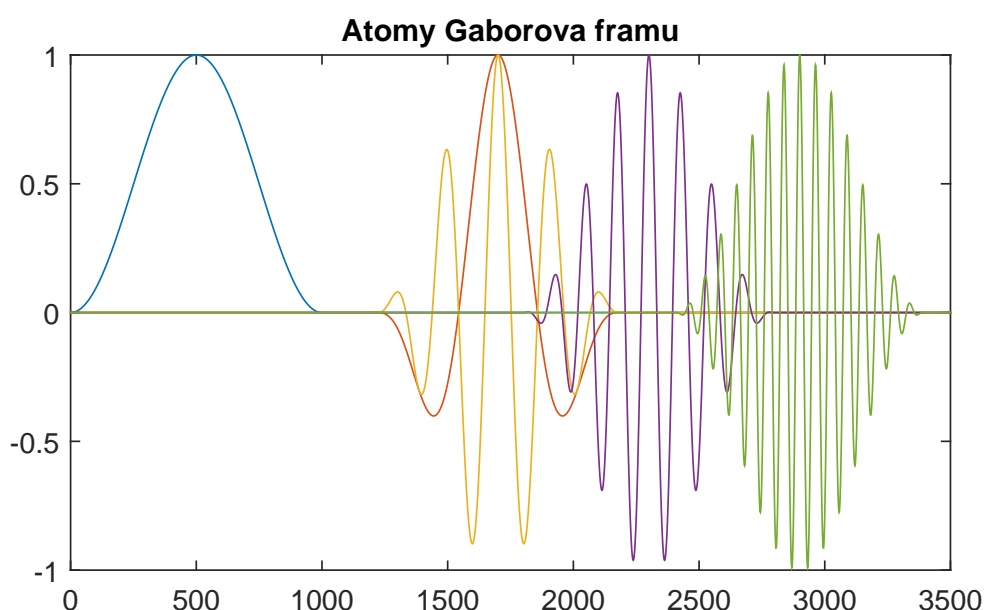
Nejjednodušší způsob dosažení časové lokalizace je vyříznutí úseku signálů a provedení Fourierovy transformace. Tento způsob ale není ideální, protože vyříznutí signálu odpovídá násobení signálu obdélníkovým oknem, což ve spektrální oblasti odpovídá konvoluci. Avšak obdélníkový signál má ve spektru funkci sinc, která pouze

pomalu klesá směrem k vysokým kmitočtům a časově-kmitočtová lokalizace tedy není ideální. Proto se v Gaborově analýze používá např. Gaussovo okno, Hannovo okno, atd.

Gaborova analýza v $L^2(\mathbb{R})$ je postavena na operátorech translace a modulace. Reprezentace funkce $f \in L^2(\mathbb{R})$ je pak superpozicí posunutých a modulovaných verzí fixní funkce $g \in L^2(\mathbb{R})$, nazvané také jako okno transformace [33]. Soubor takovýchto funkcí:

$$\left\{ e^{j2\pi mbx} g(x - na) \right\}_{m,n \in \mathbb{Z}} \quad (4.4)$$

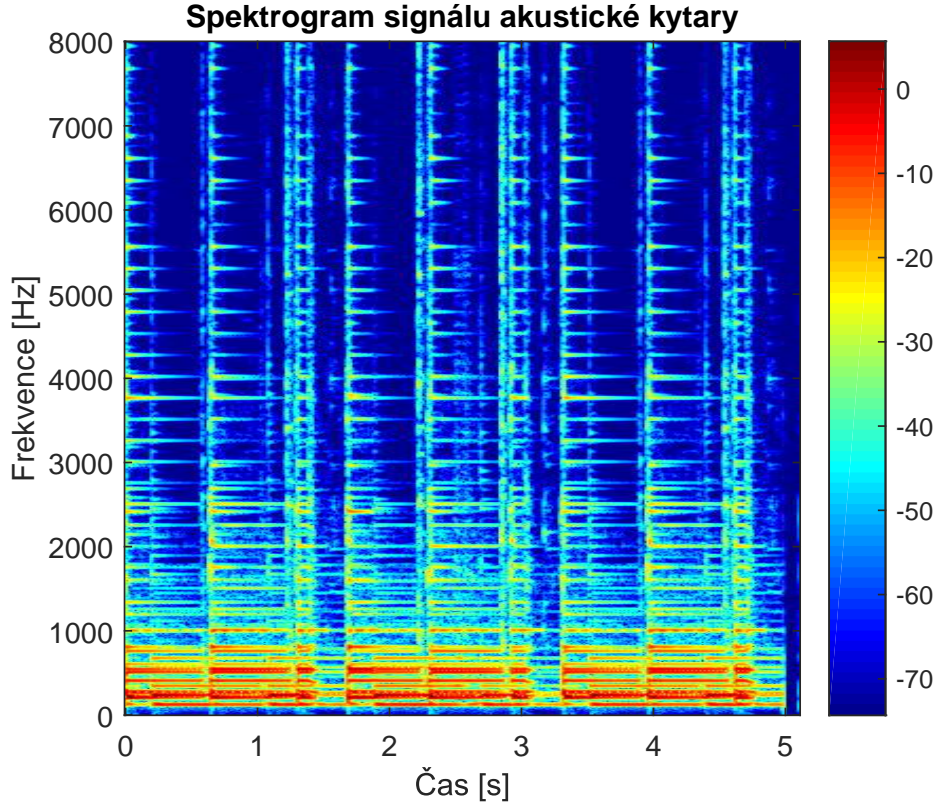
pro zvolené hodnoty a a b se nazývá *Gaborův systém*. Můžeme se také setkat s označením krátkodobá Fourierova transformace (*Short Time Fourier Transform* – SFTF).



Obr. 4.1: Ukázka několika atomů Gaborova framu s Hannovým oknem, zobrazena pouze reálná část.

Pro zobrazení gaborovských koeficientů se často používá tzv. spektrogram, což je dvourozměrný graf, kde vodorovná osa zobrazuje čas, svislá osa kmitočty a barevná škála potom reprezentuje velikost (modul) koeficientu. Příklad spektrogramu pro krátký zvukový signál akustické kytary lze vidět na obrázku 4.2.

Při práci s Gaborovou transformací se snažíme o to, aby Gaborův systém (4.4) se zvolenou funkcí $g \in L^2(\mathbb{R})$ a parametry translace a modulace a, b tvořil frame v prostoru $L^2(\mathbb{R})$. Dodnes není vyčerpávajícím způsobem vyřešeno, za jakých podmínek systém tvoří frame, avšak existuje alespoň nutná podmínka k existenci Gaborova framu. Ta říká, že pokud $a, b > 0$ a $g \in L^2(\mathbb{R})$, pak gaborovský systém (4.4) může být frame pro $L^2(\mathbb{R})$, pouze pokud platí $ab \leq 1$. Jinými slovy nutnou podmínkou pro



Obr. 4.2: Spektrogram zvukového signálu akustické kytary.

gaborovský frame je, aby časově-kmitočtová mřížka byla dostatečně hustá. Není to však podmínka dostačující. Pokud je gaborovský systém frame, pak je možné každý signál rekonstruovat z jeho časově-kmitočtových koeficientů [33].

4.5 Constant-Q transformace

V předchozí části byla představena Gaborova transformace, která má pro zpracování zvuku oproti klasické Fourierově transformaci několik výhod. Další možností je pak použít Constant-Q transformaci (CQT), která přináší další výhody oproti DGT. Srovnání CQT a DGT bude diskutováno v závěru této části.

Constant-Q transformace převádí signál z časové oblasti do kmitočtové oblasti, podobně jako Fourierova transformace nebo DGT. K tomu však používá banku logaritmicky rozmístěných filtrů, pro jejichž centrální frekvence platí vztah [34]:

$$f_k = 2^{\frac{1}{b}} f_{k-1} = 2^{\frac{k}{b}} f_0, \quad (4.5)$$

kde b značí počet filtrů (neboli *binů*) na oktávu. Šířka filtrů Δ_k je pak nejčastěji určena jako

$$\Delta_k = f_{k+1} - f_k = f_k(2^{\frac{1}{b}} - 1). \quad (4.6)$$

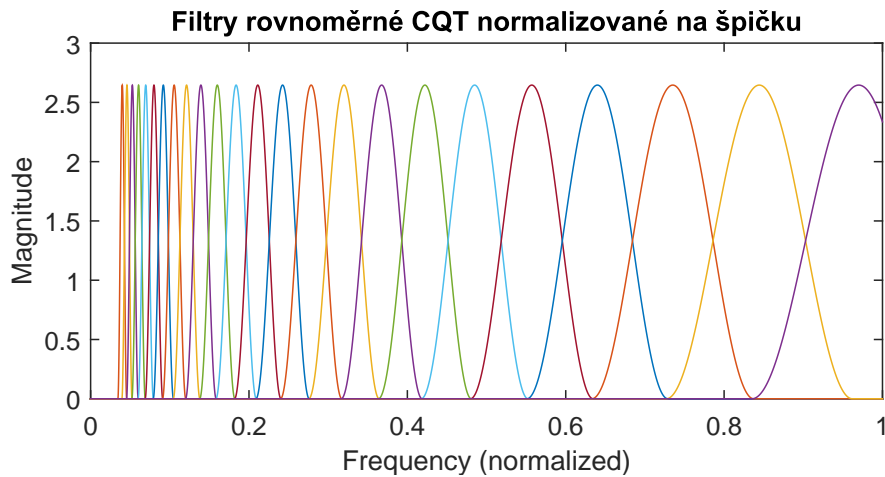
Činitel jakosti Q pak definujeme jako poměr centrální frekvence filtru ku šířce filtru, formálně tedy:

$$Q = \frac{f_k}{\Delta_k} = \frac{2^{\frac{k}{b}} f_0}{2^{\frac{k}{b}} f_0 (2^{\frac{1}{b}} - 1)} = \frac{1}{2^{\frac{1}{b}} - 1}. \quad (4.7)$$

Ze vztahu 4.7 je patrné, že činitel jakosti Q je pro všechny filtry konstantní – odtud název Constant-Q transformace.

Protože maximální kmitočet signálu po filtraci v jednotlivých pásmech je mnohem menší než maximální kmitočet celého signálu, dochází zde ve většině případů k podvzorkování signálů v jednotlivých subpásmech. Podle koeficientu podvzorkování pak dělíme banky filtrů na rovnoměrné (*uniform*) a nerovnoměrné (*non-uniform*). Rovnoměrné banky filtrů mají konstantní koeficient podvzorkování pro všechny filtry a všechny filtry tedy mají stejná maxima modulové kmitočtové charakteristiky. Takovéto banky filtrů však produkují nejvíce redundantní systémy. Naopak nerovnoměrné banky filtrů, kde je každé subpásmo podvzorkováno co nejvíce, vytváří nejméně redundantní systémy. V praxi se pak používají i různé kombinace, např. nerovnoměrné banky filtrů, u kterých jsou koeficienty podvzorkování konstantní v oktávách, ale mezi oktávami se liší [35].

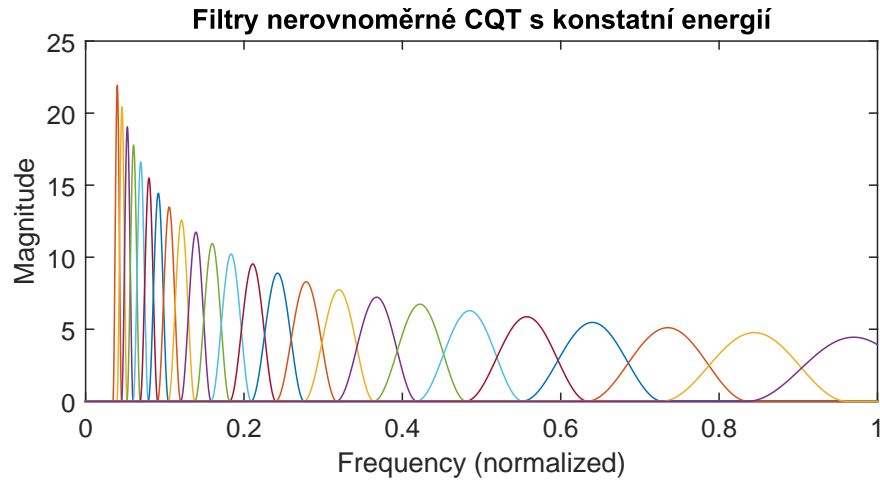
Na obrázku 4.3 pak můžeme vidět modulové kmitočtové charakteristiky uniformní banky filtrů. Protože se jedná o uniformní banku filtrů, koeficient podvzorkování je pro všechna pásma stejný. Z grafu je pak patrné, že maximální modulová kmitočtová charakteristika je u všech filtrů stejná. To ale se vzrůstající šířkou pásma jednotlivých filtrů způsobuje, že energie obsažená v impulsních charakteristikách filtrů s vyššími frekvencemi je obecně vyšší, což je v některých případech nežádoucí efekt.



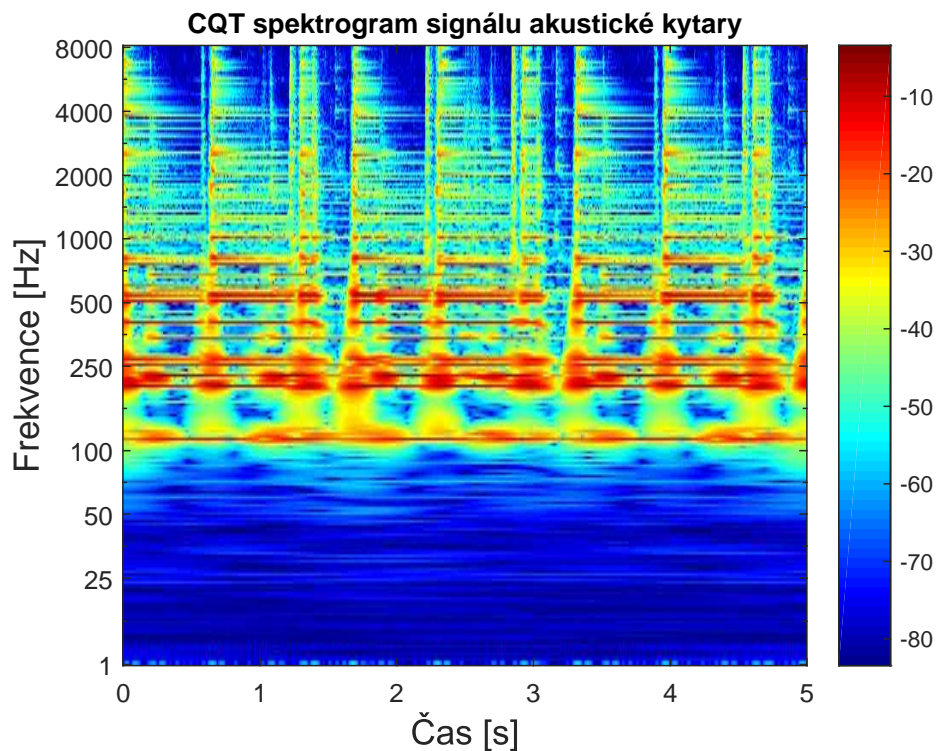
Obr. 4.3: Modulové kmitočtové charakteristiky filtrů rovnoměrné (uniformní) CQT.

Naopak na obrázku 4.4 lze vidět modulové kmitočtové charakteristiky nerovnoměrně podvzorkované banky filtrů. Každé pásmo zde bylo podvzorkováno jiným ko-

eficientem podvzorkování, a proto maximální modulová kmitočtová charakteristika jednotlivých filtrů klesá směrem k vyšším frekvencím. Zde je koeficient podvzorkování zvolen tak, aby energie obsažená v jednotlivých impulsních odezvách filtrů byla stejná.



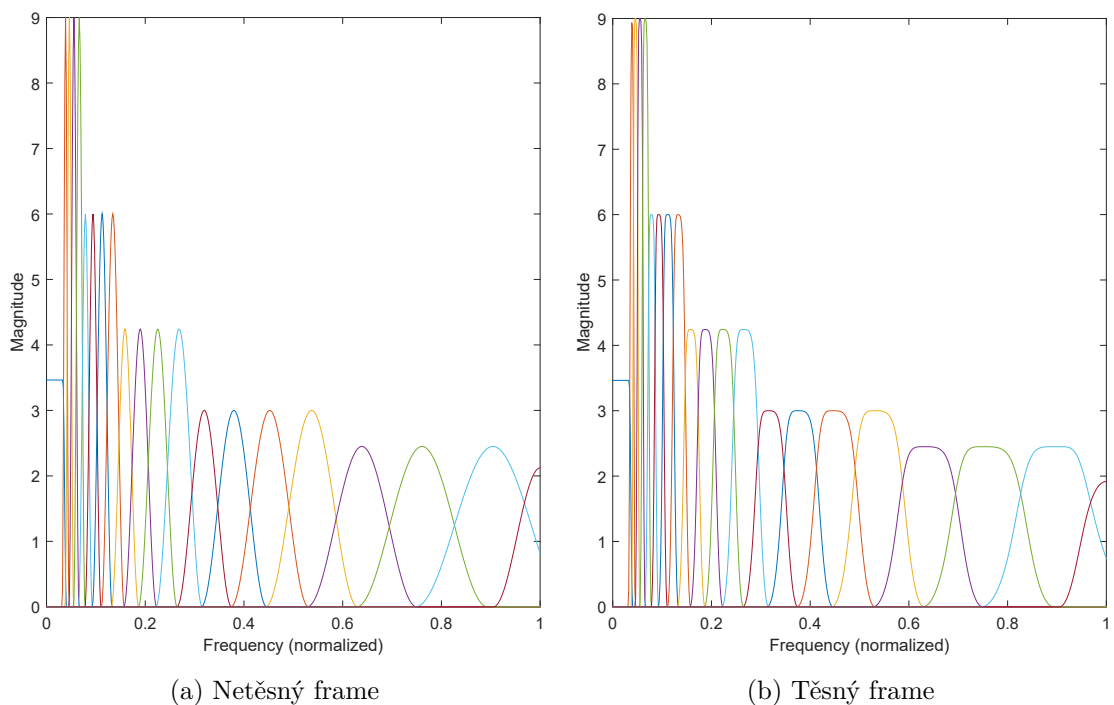
Obr. 4.4: Modulové kmitočtové charakteristiky filtrů nerovnoměrné CQT s konstantními ℓ_2 -normami.



Obr. 4.5: CQT spektrogram signálu z akustické kytary.

Pokud srovnáme CQT s DGT, pak můžeme konstatovat, že DGT má konstantní časové a frekvenční rozlišení a jednotlivá okna jsou od sebe lineárně vzdálena. V některých aplikacích může toto chování být žádoucí, ale zcela neodpovídá lidskému modelu slyšení. Naopak geometrické rozložení filtrů a konstantní činitel jakosti Q u CQT banky filtrů způsobí, že časové rozlišení se zvyšuje směrem k vyšším frekvencím. Jinými slovy nízké frekvence mají lepší frekvenční rozlišení, ale horší časové rozlišení a naopak vyšší frekvence mají lepší časové rozlišení, což mnohem lépe odpovídá tomu, jak funguje lidské slyšení [36]. Dále pokud si zvolíme základní frekvenci f_0 a 12, resp. 24 filtrů na oktávu, pak jednotlivé filtry budou přímo odpovídat hudebním notám (tónům, resp. půltónům). Tohoto faktu se pak s výhodou využívá např. při automatickém rozpoznávání hudebních nástrojů nebo transkripci skladeb.

Na obrázku 4.5 pak lze vidět spektrogram signálu akustické kytary (srov. se spektrogramem 4.2) pro uniformní CQT s 24 filtry na oktávu. Při pohledu na tento spektrogram je patrný vliv konstantního činitele jakosti Q , který způsobuje vyšší kmitočtové rozlišení nízkofrekvenčních složek a naopak lepší časové rozlišení vysokofrekvenčních složek.



Obr. 4.6: Závislost modulových kmitočtových charakteristik CQT banky filtrů na těsnosti framu.

Podobně jako u Gaborovy transformace, i pro CQT je žádoucí, aby systém v prostoru $L^2(\mathbb{R})$ tvořil frame. Tedy pokud modulové kmitočtové charakteristiky jednotlivých filtrů (atomy) poskládáme po sloupcích do matice, která splňuje podmínku

4.3, tvoří tento systém v prostoru $L^2(\mathbb{R})$ frame. V určitých aplikacích je vhodné, aby byl tento frame těsný, popř. dokonce parsevalovsky těsný (frame operátor \mathbf{FF}^* je identita). V tomto případě pak dojde k mírné deformaci modulových kmitočtových charakteristik jednotlivých filtrů. Tento jev je znázorněn na obrázku 4.6. Pro ilustraci zde byla použita neuniformní banka filtrů CQT, která má však v rámci oktávy konstantní koeficient podvzorkování. V rámci lepší přehlednosti byly použity pouze čtyři filtry na oktávu.

Při práci se CQT framem může nastat problém, kdy pro určitou aplikaci potřebujeme použít těsný frame uniformní CQT, zároveň ale chceme, aby ℓ_2 -normy modulových kmitočtových charakteristik jednotlivých filtrů (energie filtrů) byly konstantní. Tyto dvě vlastnosti se bohužel navzájem vylučují. Tento problém však lze vyřešit váhováním koeficientů, kdy použijeme těsný frame uniformní CQT, avšak jsme schopni také zkonstruovat váhy jednotlivých filtrů podle jejich energie a těmito váhami pak násobit jednotlivé koeficienty.

5 PROXIMÁLNÍ ALGORITMY A JEJICH VY- UŽITÍ PRO ÚLOHU DECLIPPINGU

5.1 Matematická formulace úlohy declipping

Mějme signál \mathbf{y} , který představuje původní, nijak nepoškozený signál. Dále uvažujme signál \mathbf{y}_c o kterém víme, že byl nějakým způsobem tvrdě ořezán podle vztahu (1.1). Budeme uvažovat případ, kdy clipping proběhl symetricky; to znamená, že prahové úrovně jsou stejně daleko od nuly. Pro přehlednost si však označíme horní práh jako $\theta_H > 0$ a dolní práh jako $\theta_L < 0$. Dále budeme předpokládat, že signál nebyl poškozen jiným způsobem, to znamená, že vzorky, které nebyly saturované, se shodují se vzorky původního signálu \mathbf{y} .

Za předpokladu, že dokážeme snadno identifikovat saturované vzorky signálu, můžeme stanovit tzv. masky signálu. V případě digitálního zpracování signálu si můžeme masku představit jako vektor logických jedniček a nul, který je stejně dlouhý, jako zpracováváný signál. Masky signálu potom slouží k výběru jednotlivých vzorků signálu, se kterými chceme aktuálně pracovat.

V našem případě si vytvoříme tři masky, které budeme při declippingu využívat. První masku si označíme jako \mathbf{M}_R a budeme jí označovat vzorky signálu, u kterých nedošlo ke clippingu – tzv. spolehlivé (*reliable*) vzorky. Další dvě masky pak budou označeny jako \mathbf{M}_H (*higher*), resp. \mathbf{M}_L (*lower*) a budou vybírat vzorky signálu, u kterých došlo k přichycení na horní úrovni θ_H , resp. dolní úrovni θ_L . Masky si lze tedy představit jako matice, které vzniknou z jednotkové matice tak, že odstraníme řádky matice, které nepatří do výběru.

Pokud tedy budeme předpokládat, že zpracováváný signál je řídký v nějaké transformaci (budeme uvažovat například tzv. Gaborovu transformaci (viz část 4.4), jinými zdroji nazývanou také jako STFT – *Short Time Fourier Transform*), rekonstruujeme saturovaný signál \mathbf{y}_c tak, že se pokusíme nalézt nejřidší vektor gaborovských koeficientů \mathbf{c} . Pro takový signál samozřejmě musí platit, že vzorky na spolehlivých pozicích budou shodné se zpracováváním signálem \mathbf{y}_c . Dále pak vzorky přichycené na práh θ_H musí ležet nad tímto prahem a samozřejmě vzorky přichycené na spodní úroveň θ_L musí ležet pod ní. Pro splnění těchto podmínek můžeme právě s výhodou použít masky signálu, které byly definovány výše.

Z části 3.3 také víme, že při hledání řídkého řešení můžeme nekonvexní ℓ_0 -normu vektoru nahradit nejbližší konvexní ℓ_1 -normou. Celou úlohu declippingu na základě

řídke reprezentace pak můžeme zapsat následovně:

$$\hat{\mathbf{c}} = \arg \min_{\mathbf{c}} \|\mathbf{c}\|_1 \quad \text{vzhledem k} \quad \begin{cases} \mathbf{M}_R G \mathbf{c} = \mathbf{M}_R \mathbf{y}_c, \\ \mathbf{M}_H G \mathbf{c} \geq \theta_H, \\ \mathbf{M}_L G \mathbf{c} \leq \theta_L, \end{cases} \quad (5.1)$$

kde G značí syntézu signálu (zpětnou Gaborovu transformaci). Po nalezení řešení problému (5.1) potom získáme rekonstruovaný signál zpětnou transformací $G\hat{\mathbf{c}}$.

5.2 Proximální algoritmy

Úloha declippingu (5.1) definovaná v předchozí části je matematicky správně, nicméně v tomto tvaru neexistuje žádný známý způsob, jak řešení efektivně nalézt. Pokud však tuto úlohu převedeme na jiný tvar, můžeme s výhodou použít nějakého zástupce z rodiny algoritmů proximálního dělení (*proximal splitting*). V této části nejprve objasním princip algoritmů proximálního dělení, popíšu tzv. proximální operátory a uvedu dva hlavní zástupce proximálních algoritmů.

5.2.1 Proximální operátory

Proximální operátor konvexní funkce f je pro každé $\mathbf{x} \in \mathbb{R}^N$ definován jako řešení minimalizační úlohy [37]:

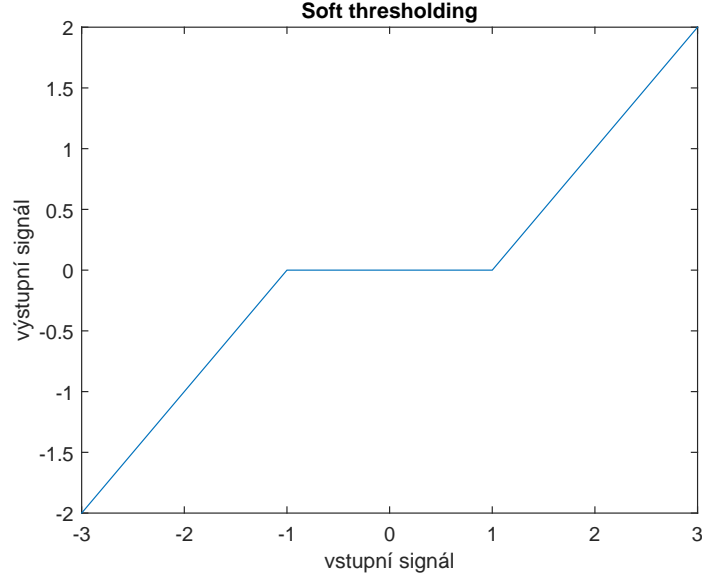
$$\text{prox}_f(\mathbf{x}) = \arg \min_{\mathbf{y} \in \mathbb{R}^N} f(\mathbf{y}) + \frac{1}{2} \|\mathbf{x} - \mathbf{y}\|_2^2. \quad (5.2)$$

Výhodou proximálních operátorů je fakt, že nám umožňují pracovat i s nediferencovatelnými funkcemi. Díky použití proximálních operátorů pak můžeme pracovat i s algoritmy, kde bychom jinak byli omezeni pouze na diferencovatelné funkce.

Z dalších částí práce vyplýne, že pro nás nejdůležitější proximální operátory budou pro ℓ_1 -normu vektoru a pro indikátorovou funkci množiny ι_C . Proximálním operátorem pro ℓ_1 -normu vektoru \mathbf{x} , neboli $\|\mathbf{x}\|_1$, je tzv. měkké prahování (*soft thresholding*), které je definováno následovně [37]:

$$\text{soft}(\mathbf{x}) = \begin{cases} \mathbf{x} - \underline{\omega} & \text{pokud } \mathbf{x} < \underline{\omega}, \\ 0 & \text{pokud } \mathbf{x} \in \langle \underline{\omega}, \bar{\omega} \rangle, \\ \mathbf{x} - \bar{\omega} & \text{pokud } \mathbf{x} > \bar{\omega}. \end{cases} \quad (5.3)$$

Pokud je prahování symetrické, lze zápis zjednodušit na $\text{soft}_\omega(\mathbf{x})$. Na grafu 5.1 je pak znázorněna převodní charakteristika systému s prahem $\omega = 1$.



Obr. 5.1: Převodní charakteristika měkkého prahování.

Indikátorová funkce ι_C je funkce, která nabývá hodnoty 0, pokud argument funkce spadá do přípustné množiny C a hodnoty $+\infty$, pokud argument leží mimo přípustnou množinu, což můžeme formálně zapsat jako

$$\iota_C(\mathbf{x}) = \begin{cases} 0 & \text{pokud } \mathbf{x} \in C, \\ +\infty & \text{pokud } \mathbf{x} \notin C, \end{cases} \quad (5.4)$$

kde $C \subset \mathbb{R}^N$ [37]. Proximálním operátorem indikátorové funkce ι_C ($\text{prox}_{\iota_C} \mathbf{x}$) je pak projekce na množinu C , což budeme zapisovat $\text{proj}_C \mathbf{x}$. Projekci na množinu C si můžeme představit jako zobrazení, které posune $\mathbf{x} \notin C$ na nejbližší bod z C a naopak $\mathbf{x} \in C$ ponechá beze změny.

5.2.2 Algoritmy proximálního dělení

Algoritmy z rodiny proximálního dělení jsou založeny na myšlence, že se velký počet optimalizačních problémů dá zapsat v tzv. neomezeném tvaru (*unconstrained form*), kdy minimalizujeme součet m konvexních funkcí f_1, \dots, f_m . V základním tvaru vypadá neomezený tvar následovně [37]:

$$\min_{\mathbf{x} \in \mathbb{R}^N} f_1(\mathbf{x}) + \dots + f_m(\mathbf{x}). \quad (5.5)$$

Pro úlohu declippingu se nabízí využít několik možností. Tou první je použít relativně prostý dopředně-zpětný (*forward-backward*) algoritmus, který řeší úlohu [37]:

$$\min_{\mathbf{x} \in \mathbb{R}^N} f_1(\mathbf{x}) + f_2(\mathbf{x}), \quad (5.6)$$

kdy obě funkce f_1 a f_2 jsou konvexní. Navíc platí, že jedna z těchto funkcí je diferencovatelná s vlastností tzv. Lipschitzova spojitého gradientu ∇f_2 . Funkce je lipschitzovsky spojitá, pokud platí [38]:

$$\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{x}')\| \leq \beta \|\mathbf{x} - \mathbf{x}'\|, \forall \mathbf{x}, \mathbf{x}' \in \mathbb{R}^N, \quad (5.7)$$

kde $\beta > 0$ se nazývá Lipschitzova konstanta.

Za předpokladu, že funkce splňují výše uvedené podmínky, lze nalézt minimálně jedno řešení pro $\gamma \in (0, +\infty)$, které charakterizuje rovnice [37]:

$$\mathbf{x} = \underset{\gamma f_1}{\text{prox}}(\mathbf{x} - \gamma \nabla f_2(\mathbf{x})). \quad (5.8)$$

Pokud budeme výše uvedený postup iterativně opakovat, je dokázáno, že algoritmus bude konvergovat k přesnému řešení. Iterativní postup můžeme zapsat následovně [37]:

$$\mathbf{x}_{n+1} = \underbrace{\underset{\gamma_n f_1}{\text{prox}}}_{\text{zpětný krok}} \left(\underbrace{\mathbf{x}_n - \gamma_n \nabla f_2(\mathbf{x}_n)}_{\text{dopředný krok}} \right). \quad (5.9)$$

Nevýhodou tohoto algoritmu je jeho omezenost, kdy potřebujeme jednu z funkcí diferencovatelnou. Můžeme však nahradit podmínku diferencovatelnosti proximálním operátorem. Takovýto algoritmus je nazýván Douglas-Rachfordův a jeho řešení je pro $\gamma \in (0, +\infty)$ charakterizováno těmito dvěma podmínkami [37]:

$$\begin{aligned} \mathbf{x} &= \text{prox}_{\gamma f_2} \mathbf{y} \\ \text{prox}_{\gamma f_2} \mathbf{y} &= \text{prox}_{\gamma f_1} (2 \text{prox}_{\gamma f_2} \mathbf{y} - \mathbf{y}). \end{aligned} \quad (5.10)$$

Zobecnění dopředně-zpětného algoritmu však může mít za následek pomalejší konvergenci algoritmu, protože se místo jednoho proximálního operátoru musejí počítat dva.

Pro úlohu zadanou třemi omezujícími podmínkami však není ani Douglas-Rachfordův dostatečně obecný. Nicméně lze však ukázat, že oba tyto algoritmy jsou pouze speciálními případy obecného proximálního algoritmu pro konvexní optimalizaci (nebo také Condatova algoritmu) [38]. Tento obecný algoritmus řeší minimalizační úlohu:

$$\arg \min_{\mathbf{x} \in \mathbb{R}^N} f(\mathbf{x}) + g(\mathbf{x}) + \sum_{m=1}^M h_m(L_m \mathbf{x}), \quad (5.11)$$

kde funkce f je diferencovatelná s Lipschitzovým spojitým gradientem. Jednotlivé funkce musejí samozřejmě splňovat určité matematické podmínky, aby byla zaručena konvergence algoritmu. Pro účely zpracování signálu však všechny tyto podmínky jsou splněny a nebudeme se jimi dále zabývat.

Celý obecný proximální algoritmus vypadá následovně [38]:

Zvolíme parametry $\tau > 0$, $\sigma > 0$, $\rho > 0$.

Zvolíme počáteční odhady $\mathbf{x}^{(0)}, \mathbf{u}_1^{(0)}, \dots, \mathbf{u}_M^{(0)}$.

For $i = 0, 1, \dots$

$$\left[\begin{array}{l} \tilde{\mathbf{x}}^{(i+1)} := \text{prox}_{\tau g} \left(\mathbf{x}^{(i)} - \tau \nabla f(\mathbf{x}^{(i)}) - \tau \sum_{m=1}^M L_m^* \mathbf{u}_m^{(i)} \right), \\ \mathbf{x}^{(i+1)} := \rho \tilde{\mathbf{x}}^{(i+1)} + (1 - \rho) \mathbf{x}^{(i)}, \\ \text{For } m = 1, \dots, M, \\ \left[\begin{array}{l} \tilde{\mathbf{u}}_m^{(i+1)} := \text{prox}_{\sigma h_m^*} \left(\mathbf{u}_m^{(i)} + \sigma L_m (2\tilde{\mathbf{x}}^{(i+1)} - \mathbf{x}^{(i)}) \right), \\ \mathbf{u}_m^{(i+1)} := \rho \tilde{\mathbf{u}}_m^{(i+1)} + (1 - \rho) \mathbf{u}_m^{(i)}. \end{array} \right. \end{array} \right. \quad (5.12)$$

V tomto algoritmu označení h_m^* představuje Fenchel-Rockafellarovo sdružení k funkci h_m a pro reálné $\sigma > 0$ pak pro proximální operátory těchto funkcí platí [38]:

$$\text{prox}_{\sigma h_m^*}(\mathbf{u}) = \mathbf{u} - \sigma \text{prox}_{h_m/\sigma}(\mathbf{u}/\sigma). \quad (5.13)$$

Konvergence tohoto algoritmu je pak zaručená (důkaz proveden v článku [39]), pokud parametry splňují následující podmínky [38]:

$$\begin{aligned} \text{(i)} \quad & \tau \left(\frac{\beta}{2} + \sigma \left\| \sum_{m=1}^M L_m^* L_m \right\| \right) < 1 \\ \text{(ii)} \quad & \rho \in (0, 1), \end{aligned} \quad (5.14)$$

kde $\|\cdot\|$ je operátorová norma a β je Lipschitzova konstanta definovaná v (5.7).

Výhodou tohoto algoritmu je i možnost některé funkce ze vztahu (5.11) považovat za nulové, což, jak bude vidět později, bude i v našem případě pro declipping. Zároveň také některé lineární operátory L_m můžeme nahradit identitami. V případě, že $f = 0$ a prostory, v nichž pracujeme, mají konečnou dimenzi, jsou konvergenční podmínky pro parametry následující [38]:

$$\begin{aligned} \text{(i)} \quad & \tau \sigma \left\| \sum_{m=1}^M L_m^* L_m \right\| \leq 1, \\ \text{(ii)} \quad & \rho \in (0, 2). \end{aligned} \quad (5.15)$$

5.3 Řešení pomocí Condatova algoritmu

Z předchozí části, ve které byly definovány proximální algoritmy, vyplývá, že pro potřeby declippingu se hodí nejlépe obecný proximální algoritmus pro konvexní optimalizaci (Condatův algoritmus). V této části bude proto základní úloha declippingu převedena na takový tvar, který bude pro Condatův algoritmus uchopitelný. Následně je popsán samotný declipping využívající Condatův algoritmus. V poslední části jsou popsány projekce, které tento algoritmus využívá.

5.3.1 Declipping pomocí Condatova algoritmu

Na začátku předchozí části bylo zmíněno, že pro vyřešení minimalizační úlohy (5.1) pomocí proximálního dělení musíme převést úlohu na jiný tvar. Tento tvar se nazývá neomezený a převádí omezující podmínky z původní úlohy (5.1) na součet m konvexních funkcí.

Pokud bychom si označili množinu C jako množinu všech přípustných řešení, která splňuje omezující podmínky úlohy (5.1), mohli bychom v neomezeném tvaru použít indikátorovou funkci ι_C definovanou v části 5.2.1. Neomezený tvar by potom vypadal následovně:

$$\hat{\mathbf{c}} = \arg \min_{\mathbf{c}} \|\mathbf{c}\|_1 + \iota_C(\mathbf{c}). \quad (5.16)$$

Tento zápis je v praxi velmi elegantní. Pokud totiž najdeme řešení, které spadá do množiny přípustných řešení C , indikátorová funkce bude rovna nule a budeme řešit pouze minimalizaci ℓ_1 normy vektoru \mathbf{c} . Naopak pokud řešení bude ležet mimo množinu C , ι_C bude rovna $+\infty$, potom všechny ostatní členy budou zanedbatelné, a proto nebude toto řešení zaručeně minimální.

Obě použité funkce jsou konvexní a nejsou diferencovatelné. To znamená, že bychom pro vyřešení této úlohy mohli použít Douglas-Rachfordův algoritmus, který přesně takovéto funkce předpokládá. Proximálními operátory jednotlivých funkcí by pak bylo měkké prahování pro ℓ_1 -normu a projekce na množinu C pro indikátorovou funkci.

Při použití vztahu 5.16 je však problémem projekce na množinu zadanou třemi podmínkami, která není triviální. Pokud se podíváme na úlohu, kterou řeší obecný proximální algoritmus (5.11), je patrné, že zde můžeme použít i sumu M konvexních funkcí h_m . Toto řešení se na první pohled jeví jako optimální. Proto množinu přípustných řešení C pojmem jako průnik třech množin, které si označíme např. R , H a L . Množina R (*reliable*) zde bude představovat taková řešení, u kterých bude platit první podmínka z úlohy (5.1), tedy $\mathbf{M}_R \mathbf{G} \mathbf{c} = \mathbf{M}_R \mathbf{y}_c$. Jinými slovy na pozicích společných vzorků dojde ke shodě. Obdobně pak množina H bude splňovat druhou podmínku pro vzorky shora satureované na úrovni θ_H a množina L pak podmínku pro vzorky zdola satureované na úrovni θ_L . Označení přípustných množin koresponduje s označením jednotlivých indikátorových funkcí, tedy ι_R, ι_H a ι_L .

Optimalizační úloha bude mít za těchto podmínek tvar:

$$\hat{\mathbf{c}} = \arg \min_{\mathbf{c}} \|\mathbf{c}\|_1 + \iota_R(\mathbf{c}) + \iota_H(\mathbf{G} \mathbf{c}) + \iota_L(\mathbf{G} \mathbf{c}). \quad (5.17)$$

Pokud ještě srovnáme tuto úlohu se základní úlohou, kterou řeší obecný algoritmus proximálního dělení, pak diferencovatelná funkce f je v tomto případě nulová, tedy $\nabla f = 0$. Funkce g odpovídá ℓ_1 -normě hledaného vektoru ($g = \|\cdot\|_1$) a suma funkcí h_m je suma jednotlivých indikátorových funkcí ι_R, ι_H a ι_L . Lineární operátory L_m ,

resp. L_m^* pak odpovídají syntéze, resp. analýze signálu. V případě použití DGT platí $L_1 = I$ a $L_2 = L_3 = G$.

Při inicializaci algoritmu je nutné správně nastavit koeficienty ρ, σ a τ . Protože funkce f je v tomto případě nulová, má koeficient ρ podle vztahu (5.15) ležet v intervalu $(0; 2)$. Ke stanovení koeficientů τ a σ použijeme první část vztahu (5.15). Nejprve je nutné určit sumu norem operátorů L_m . V našem konkrétním případě používáme pro množinu R operátor identity a pro množiny H a L operátor Gaborovy transformace G . Protože pracujeme s Parsevalovými framy, jejich operátorové normy budou rovny jedné. Formálně

$$\|I + G^*G + G^*G\| \leq \|I\| + \|G^*G\| + \|G^*G\| = 1 + 1 + 1 = 3. \quad (5.18)$$

Z toho vyplývá, že pro koeficienty τ a σ musí platit vztah

$$\tau \cdot \sigma \cdot 3 \leq 1 \quad \text{neboli} \quad \tau \leq \frac{1}{3\sigma}. \quad (5.19)$$

Vzájemná závislost těchto koeficientů tedy v prvním kvadrantu kartézského souřadnicového systému tvoří hyperbolu. Ve většině případů se koeficienty nastavují tak, aby ležely přímo na dané hyperbole, neostrou nerovnost ve vztahu (5.19) pak vystřídá rovnost.

Samotná aplikace obecného proximálního algoritmu pro potřeby declippingu vypadá následovně:

$$\begin{aligned} & \text{Zvolíme parametry } \rho \in (0; 2), \sigma > 0, \tau = \frac{1}{3\sigma}. \\ & \text{Zvolíme počáteční } \mathbf{c}^{(0)}, \mathbf{u}_R^{(0)}, \mathbf{u}_H^{(0)}, \mathbf{u}_L^{(0)}. \\ & \text{For } i = 0, 1, \dots \\ & \quad \left[\begin{aligned} \tilde{\mathbf{c}}^{(i+1)} &:= \text{soft}_\tau \left(\mathbf{c}^{(i)} - \tau \left(\mathbf{u}_R^{(i)} + G^* \mathbf{u}_H^{(i)} + G^* \mathbf{u}_L^{(i)} \right) \right), \\ \mathbf{c}^{(i+1)} &:= \rho \tilde{\mathbf{c}}^{(i+1)} + (1 - \rho) \mathbf{c}^{(i)}, \\ \text{Reliable:} \\ \quad \left[\begin{aligned} \alpha &:= \mathbf{u}_R^{(i)} + \sigma \left(2\tilde{\mathbf{c}}^{(i+1)} - \mathbf{c}^{(i)} \right), \\ \tilde{\mathbf{u}}_R^{(i+1)} &:= \alpha - \sigma \text{proj}_R(\alpha/\sigma), \\ \mathbf{u}_R^{(i+1)} &:= \rho \tilde{\mathbf{u}}_R^{(i+1)} + (1 - \rho) \mathbf{u}_R^{(i)}, \end{aligned} \right. \\ \text{High:} \\ \quad \left[\begin{aligned} \beta &:= \mathbf{u}_H^{(i)} + \sigma G \left(2\tilde{\mathbf{c}}^{(i+1)} - \mathbf{c}^{(i)} \right), \\ \tilde{\mathbf{u}}_H^{(i+1)} &:= \beta - \sigma \text{proj}_H(\beta/\sigma), \\ \mathbf{u}_H^{(i+1)} &:= \rho \tilde{\mathbf{u}}_H^{(i+1)} + (1 - \rho) \mathbf{u}_H^{(i)}, \end{aligned} \right. \\ \text{Low:} \\ \quad \left[\begin{aligned} \gamma &:= \mathbf{u}_L^{(i)} + \sigma G \left(2\tilde{\mathbf{c}}^{(i+1)} - \mathbf{c}^{(i)} \right), \\ \tilde{\mathbf{u}}_L^{(i+1)} &:= \gamma - \sigma \text{proj}_L(\gamma/\sigma), \\ \mathbf{u}_L^{(i+1)} &:= \rho \tilde{\mathbf{u}}_L^{(i+1)} + (1 - \rho) \mathbf{u}_L^{(i)}. \end{aligned} \right. \end{aligned} \right. \quad (5.20) \end{aligned}$$

Jak si lze z výše uvedeného algoritmu všimnout, obecné proximální operátory zde už byly nahrazeny konkrétními operátory pro danou funkci. Pro ℓ_1 -normu vektoru \mathbf{c} je to operátor měkkého prahování s prahem τ (soft_τ) a pro indikátorové funkce je zde operátor projekce na danou množinu (proj). Proximální operátory, ve kterých v obecném tvaru obecného proximálního algoritmu (5.12) vystupovala Fenchel-Rockafellarova sdružená funkce, už jsou v tomto tvaru přepočítány podle vztahu (5.13).

Protože se jedná o konvexní úlohu, volba počátečních odhadů by neměla ovlivnit nalezené řešení. Lze tedy jako počáteční odhad použít samé nuly nebo např. náhodně vygenerovaná čísla. Pravděpodobně nejlepší volbou je však zvolit jako počáteční odhady zpracováváný saturevaný signál, avšak bylo empiricky zjištěno, že volba počátečních odhadů má pouze nepatrný vliv na rychlost nalezení řešení.

5.3.2 Projekce na množinu přípustných řešení

Tato část práce bude zaměřena na operátor projekce na množinu přípustných řešení, což je jeden z hlavních kroků při řešení audio declippingu pomocí proximálního algoritmu. Obecná projekce vektoru \mathbf{x} na množinu C , tedy $\text{proj}_C(\mathbf{x})$ je definována následovně:

$$\arg \min_{\mathbf{z}} \|\mathbf{x} - \mathbf{z}\|_2 \quad \text{vzhledem k } \mathbf{z} \in C. \quad (5.21)$$

Prakticky to znamená, že minimalizujeme vzdálenost mezi \mathbf{x} a libovolným bodem $\mathbf{z} \in C$, jinými slovy hledáme takový bod, který je nejbližší původnímu \mathbf{x} , ale zároveň leží v C .

Je nutné rozlišovat, na kterou množinu se projekce provádí, protože projekce nejsou implementovány totožně. Nejprve zde bude popsána projekce na množinu R a následně projekce na množiny H a L .

Projekce na množinu R

Projekce na množinu R zajišťuje, že rekonstruovaný signál zůstane na spolehlivých vzorcích nezměněn. Mějme tedy vektor \mathbf{c} , který chceme projektovat na množinu $R = \{\mathbf{z} | \mathbf{M}_R G \mathbf{z} = \mathbf{M}_R \mathbf{y}_c\}$. Pokud G je těsný Parsevalův frame, pak můžeme projekci provést následovně [40]:

$$\text{proj}_{\{\mathbf{z} | \mathbf{M}_R G \mathbf{z} = \mathbf{M}_R \mathbf{y}_c\}}(\mathbf{c}) = \mathbf{c} - G^*((\mathbf{M}_R)^\top \mathbf{M}_R) G \mathbf{c} + G^*(\mathbf{M}_R)^\top \mathbf{M}_R \mathbf{y}_c. \quad (5.22)$$

Druhý sčítanec ve výše uvedené projekci znamená, že na vstupní koeficienty aplikujeme syntézu, abychom se dostali do oblasti signálu. Zde vynulujeme vzorky na nespolehlivých pozicích a následně provedeme zpět analýzu. Třetí sčítanec pak pracuje pouze se vstupním signálem (ve kterém vynuluje nespolehlivé vzorky a provede analýzu) a proto může být předpočítán.

Výše uvedená projekce (5.22) vychází z předpokladu, že projekci na množinu $\{\mathbf{z}|\mathbf{D}\mathbf{z} = \mathbf{y}\}$ lze provést jako [40]:

$$\text{proj}_{\{\mathbf{z}|\mathbf{D}\mathbf{z}=\mathbf{y}\}}(\mathbf{c}) = \mathbf{c} - \mathbf{D}^+(\mathbf{D}\mathbf{c} - \mathbf{y}) = (\mathbf{I} - \mathbf{D}^+\mathbf{D})\mathbf{c} + \mathbf{D}^+\mathbf{y}, \quad (5.23)$$

kde \mathbf{D} je matice a horní index $+$ značí pseudoinverzi. Pro těsný Parsevalův frame \mathbf{D} platí, že $(\mathbf{D}_R)^+ = \mathbf{D}^*(\mathbf{M}_R)^\top$, kde index R označuje vzorky signálu na spolehlivých pozicích, popř. tomu odpovídající vzorky atomů ve slovníku \mathbf{D} .

Projekce na množiny H a L

Projekce na množiny $H = \{\mathbf{z}|\mathbf{M}_H\mathbf{z} \geq \theta_H\}$ a $L = \{\mathbf{z}|\mathbf{M}_L\mathbf{z} \leq \theta_L\}$ pro rekonstruovaný signál zajišťuje splnění podmínek $\mathbf{M}_H G\mathbf{c} \geq \theta_H$ a $\mathbf{M}_L G\mathbf{c} \leq \theta_L$. Jinými slovy, cílem této projekce je zabezpečit, aby se vzorky rekonstruovaného signálu, u kterých došlo ke clippingu, nevyskytovaly pod úrovní θ_H a nad úrovní θ_L .

U projekcí na množiny H a L hrají významnou roli lineární operátory L_2 a L_3 , neboli operátory syntézy signálu. Díky těmto operátorům se pak pohybujeme v oblasti signálu, což samotnou projekci výrazně zjednodušuje. Stačí pouze identifikovat vzorky, které nesplňují výše uvedené podmínky a „posunout“ je na nejbližší bod, který leží v požadované množině H , resp. L . V praxi to znamená, že takovým vzorkům signálu přiřadíme hodnotu θ_H , resp. θ_L .

5.4 Řešení pomocí algoritmu Douglas-Rachford

V části 5.3.1 je základní úloha pro declipping převedena na neomezený tvar, aby mohla být následně vyřešena pomocí proximálního algoritmu. Tento neomezený tvar vypadá následovně:

$$\hat{\mathbf{c}} = \arg \min_{\mathbf{c}} \|\mathbf{c}\|_1 + \iota_C(\mathbf{c}), \quad (5.24)$$

kde množina C představuje množinu všech přípustných řešení splňující omezující podmínky (5.1).

V této kapitole je popsána metoda declippingu, která s použitím jistého triku pracuje přímo s tímto tvarem (5.24), na rozdíl od metody popsané v předchozí kapitole, kde byla množina C rozdělena na tři jiné množiny.

5.4.1 Definice množiny C

Množina všech přípustných řešení C již byla diskutována v části 5.3.1. Pomocí této definice však nejsme schopni na takovouto množinu provést projekci. Zavedeme si

proto dva pomocné vektory \mathbf{b}_1 a \mathbf{b}_2 , pomocí kterých vymežíme prostor množiny C . Tyto vektory definujeme následovně:

$$(\mathbf{b}_1)_i = \begin{cases} \mathbf{y}_i & \text{pokud } i \in \mathbf{M}_R, \\ \theta_H & \text{pokud } i \in \mathbf{M}_H, \\ -\infty & \text{pokud } i \in \mathbf{M}_L, \end{cases} \quad (\mathbf{b}_2)_i = \begin{cases} \mathbf{y}_i & \text{pokud } i \in \mathbf{M}_R, \\ \infty & \text{pokud } i \in \mathbf{M}_H, \\ \theta_L & \text{pokud } i \in \mathbf{M}_L, \end{cases} \quad (5.25)$$

kde i označuje i -tý vzorek signálu. Vektor \mathbf{b}_1 tedy omezuje signál „zdola“, \mathbf{b}_2 naopak „shora“. Pokud je např. j -tý vzorek signálu spolehlivý (tj. nebyl saturován), objeví se tento vzorek na pozici j ve vektorech \mathbf{b}_1 i \mathbf{b}_2 , tedy $(\mathbf{b}_1)_j = (\mathbf{b}_2)_j = \mathbf{y}_j$.

Pomocí těchto vektorů pak lze stanovit množinu $C = \{\mathbf{z} | \mathbf{b}_1 \leq G\mathbf{z} \leq \mathbf{b}_2\}$. Použitá projekce s tímto vyjádřením množiny C je rozebrána v části 5.4.3.

5.4.2 Declipping pomocí algoritmu Douglas-Rachford

Pokud budeme vycházet z tvaru (5.24), kde první funkcí je ℓ_1 -norma a druhou funkcí je indikátorová funkce ι_C , můžeme použít k vyřešení takovéto úlohy proximální algoritmus Douglas-Rachford. Tento algoritmus předpokládá dvě konvexní funkce, které nejsou diferencovatelné, což je přesně tento případ.

Obecně vypadá Douglas-Rachford algoritmus následovně [37]:

$$\begin{aligned} & \text{Zvolíme } \epsilon \in (0, 1), \gamma > 0, \mathbf{y}_0 \in \mathbb{R}^N \\ & n = 0, 1, 2, \dots \\ & \mathbf{x}_n = \text{prox}_{\gamma f_2} \mathbf{y}_n \\ & \lambda_n \in \langle \epsilon, 2 - \epsilon \rangle \\ & \mathbf{y}_{n+1} = \mathbf{y}_n + \lambda_n \left(\text{prox}_{\gamma f_1} (2\mathbf{x}_n - \mathbf{y}_n) - \mathbf{x}_n \right). \end{aligned} \quad (5.26)$$

Pokud dosadíme za jednotlivé proměnné a obecné proximální operátory nahradíme konkrétními použitými operátory, pak Douglas-Rachford algoritmus pro účely declippingu bude vypadat takto:

$$\begin{aligned} & \text{Zvolíme } \epsilon \in (0, 1), \gamma > 0, \mathbf{y}_0 \in \mathbb{R}^N \\ & n = 0, 1, 2, \dots \\ & \tilde{\mathbf{c}}_n = \text{proj}_C \mathbf{c}_n \\ & \lambda_n \in \langle \epsilon, 2 - \epsilon \rangle \\ & \mathbf{c}_{n+1} = \mathbf{c}_n + \lambda_n \left(\text{soft}_{\gamma_n} (2\tilde{\mathbf{c}}_n - \mathbf{c}_n) - \mathbf{c}_n \right). \end{aligned} \quad (5.27)$$

5.4.3 Projekce na množinu přípustných řešení

Pro účely projekce na množinu přípustných řešení C je nutné nejprve provést několik definic. Nechť n je sudé číslo a nechť \mathbb{G} představuje prostor komplexních matic [41]

$$\mathbb{G}^{m,n,N} = \{[\mathbf{B}_0 \dots \mathbf{B}_{N-1}] | \mathbf{B}_i \in \mathbb{K}^{m \times n}\}, \quad (5.28)$$

tedy matice \mathbf{B} z prostoru $\mathbb{K}^{m \times n}$ jsou vyskládány do řádku. Dále necht $\mathbb{G}^{n,N}$ představuje prostor vektorů se stejnou strukturou, tedy [41]

$$\mathbb{G}^{n,N} = \left\{ \begin{bmatrix} \mathbf{b}_0 \\ \vdots \\ \mathbf{b}_{N-1} \end{bmatrix} \mid \mathbf{b}_i \in \mathbb{K}^n \right\}. \quad (5.29)$$

Takovýto případ nastává např. v Gaborově transformaci, kdy syntezující matice patří do $\mathbb{G}^{m,n,N}$.

Dále necht $\tilde{\mathbb{R}}$ představuje rozšířenou reálnou osu, tedy $\tilde{\mathbb{R}} = \mathbb{R} \cup \{-\infty, \infty\}$. Pokud předpokládáme matici plné hodnosti $A \in \mathbb{G}^{m,n,N}$, kdy $m \leq n$ a $\mathbf{A}\mathbf{A}^*$ je diagonální a vektory $\mathbf{b}_1, \mathbf{b}_2 \in \tilde{\mathbb{R}}$ takové, že $\mathbf{b}_1 \leq \mathbf{b}_2$, pak projekci vektoru $\mathbf{z} \in \mathbb{G}^{n,N}$

$$\text{proj}_{\{\mathbf{x} \mid \mathbf{A}\mathbf{x} \in [\mathbf{b}_1, \mathbf{b}_2]\}}(\mathbf{z}) := \arg \min_{\mathbf{u}} \|\mathbf{z} - \mathbf{u}\|_2 \text{ vzhledem k } \mathbf{A}\mathbf{u} \in [\mathbf{b}_1, \mathbf{b}_2] \text{ a } \mathbf{u} \in \mathbb{G}^{n,N}$$

můžeme vyjádřit jako [41]:

$$\text{proj}_{\{\mathbf{x} \mid \mathbf{A}\mathbf{x} \in [\mathbf{b}_1, \mathbf{b}_2]\}}(\mathbf{z}) = \mathbf{z} - \mathbf{A}^+ \left(\mathbf{A}\mathbf{z} - \text{proj}_{[\mathbf{b}_1, \mathbf{b}_2]}(\mathbf{A}\mathbf{z}) \right), \quad (5.30)$$

kde uvedená projekce může být vypočtena jako [41]:

$$\text{proj}_{[\mathbf{b}_1, \mathbf{b}_2]}(\mathbf{y}) = \min(\max(\mathbf{b}_1, \mathbf{y}), \mathbf{b}_2). \quad (5.31)$$

Důkaz této projekce je nad rámec této práce, avšak je k dohledání v článku [41] Pavla Rajmice a Vítězslava Veselého.

6 PROGRAMOVÉ ŘEŠENÍ

Tato kapitola je zaměřena především na samotnou implementaci jednotlivých algoritmů pro declipping, matematicky popsanych v kapitole 5. Všechny algoritmy jsou implementovány v prostředí Matlab. Pro implementaci byla použita verze 2014b; zpětná kompatibilita se staršími verzemi tedy není zajištěna.

Pro spuštění kódů declippingu je nutné mít nainstalovaný LTFAT (*The Large Time-Frequency Analysis Toolbox*) [42], což je toolbox pro Matlab/Octave pro práci s časově-frekvenční analýzou a syntézou. LTFAT je ke stažení na webových stránkách <http://ltfat.github.io/>.

Cílem této kapitoly je zdokumentovat programové řešení declippingu. Celé řešení obsahuje několik m-souborů, které jsou v této kapitole popsány a je zde vysvětlena také jejich funkčnost a možnosti nastavení.

Program pro declipping ve stručnosti funguje následovně: po spuštění hlavního souboru je načten vstupní signál, který je normalizován. Dále jsou inicializovány různé parametry týkající se rekonstrukce signálu. Následně je vstupní signál omezen na předem nastavenou hodnotu. Poté je podle předchozího nastavení zkonstruován frame, který společně se saturovaným signálem vstupuje do zvoleného proximálního algoritmu, který nalezne požadovaný rekonstruovaný signál. Ten je následně vykreslen, přehrán a do konzoly jsou vypsány informace o časovém trvání algoritmu a o hodnotách SNR.

6.1 Hlavní soubor

Hlavním programem balíku pro declipping je dávkový soubor `declipping_main.m`, který je rozdělen do šesti hlavních částí.

Načtení vstupního zvukového souboru

V první části probíhá výběr zvukového signálu ke zpracování. Program ve výchozím nastavení předpokládá, že zvukové soubory se budou nacházet ve složce `sounds`, avšak toto nastavení lze provést změnou hodnoty proměnné `audio_path`. Konkrétní výběr zvukového souboru pak obstarává proměnná `audio_file`.

Pro načtení zvukového souboru pak slouží funkce `audioread`, která od verze 2012b nahrazuje starší funkci `wavread`. Výstupem této funkce je v případě mono signálu vektor vzorků zvukového signálu v hodnotách od -1 do 1 a hodnota vzorkovací frekvence.

Při zpracování signálů bylo vždy počítáno se zvukovým formátem WAVE (*.wav), avšak funkce `audioread` si poradí i s formáty typu OGG, FLAC, AU, AIFF, AIFC,

MP3 i MPEG-4 AAC. Důležité však je, aby zvukový signál byl jednobandový, protože aplikace pro declipping zatím neumí pracovat s vícebandovými soubory.

Posledním krokem v této části je normalizace vstupního signálu. Maximální velikost absolutní hodnoty signálu tak bude jednička.

Nastavení

Tato část slouží především k nastavení různých parametrů celého programu. Všechny parametry jsou obsaženy ve struktuře `param`, která v případě nutnosti vstupuje do jednotlivých funkcí.

Jako první se v programu vyskytuje proměnná `theta`, která obsahuje práh tvrdého clippingu θ_c ze vztahu (1.1).

Dále následuje nastavení použitého framu. Proměnnou `frametype` typu string lze nastavit na hodnotu `'dgtreal'` pro Gaborův frame (viz část 4.4) nebo `'cqt'` pro Constant-Q frame (viz část 4.5). Pro frame typu DGT je pak dále potřeba nastavit šířku okna `w`, posun okna `a`, počet kmitočtových kanálů `M` a typ okna `wtype`. Šířku a posun okna lze nastavit ručně, ve výchozím nastavení ji však obstarává funkce `win_param` (viz část 6.2). Pro frame typu CQT je pak nutné nastavit typ banky filtrů `cqtType`, minimální frekvenci `fmin`, maximální frekvenci `fmax` a počet filtrů na oktávu `bins`.

Při zpracování dlouhých signálů můžeme přepínačem `segmentation` aktivovat zpracování signálu po blocích (segmentech). V tomto případě pak lze nastavit délku segmentu ve vzorcích v proměnné `seg` a velikost překryvu jednotlivých segmentů také ve vzorcích v proměnné `overlap`. Ve výchozím nastavení je tato možnost vypnutá.

Proměnnou `algorithm` lze nastavit použitý proximální algoritmus. Možnosti jsou `'Condat'` pro declipping podle části 5.3.1 nebo `'Douglas-Rachford'` podle části 5.4.2.

Přepínač `replaceReliable` slouží k nahrazení vzorků na spolehlivých pozicích původními vzorky ze vstupního saturevaného signálu (po dokončení všech iterací proximálního algoritmu). Tento přepínač nemá smysl pro Douglas-Rachford algoritmus, protože v něm je použita projekce natolik přesná, že odchylka na spolehlivých vzorcích je minimální (za předpokladu, že posledním krokem, který algoritmus vykonává, je projekce). Avšak pro obecný proximální algoritmus (Condat), kde se výsledná projekce počítá jako součet třech různých projekcí, může být odchylka na spolehlivých vzorcích znatelná, zejména při malém počtu iterací.

Parametr `reweighting` pak funguje pouze pro CQT a umožňuje váhování koeficientů z důvodů popsanych v posledním odstavci části 4.5. S váhováním koeficientů souvisí i proměnná `weightCoef`, která provádí umocňování těchto vah. Výsledné

váhy se následně vypočítají jako $\text{weights} = 1/\text{scal}^{\text{weightCoef}}$. Čím větší hodnoty tato proměnná dosahuje, tím více jsou potlačeny vysokofrekvenční složky signálu. Empiricky se totiž potvrzuje, že mírná penalizace vysokofrekvenčních složek přispívá ke kvalitnější rekonstrukci. Nejlepší výsledky v SNR podává algoritmus pro $\text{weightCoef} = 4$, což je i výchozí hodnota této proměnné. Při tomto nastavení se však penalizace vysokofrekvenčních složek projeví i perceptuálně (negativně).

Poslední parametr v části nastavení se nazývá `plotfreqz` a pokud je nenulový, program zobrazí kmitočtovou charakteristiku banky filtrů v případě, že je zvolena CQT. Pro DGT nemá tento parametr význam.

Clipping

V této části programu se volá funkce `hard_clip.m` (viz část 6.3), která na základě vstupního signálu a úrovní clippingu $-\theta_c$ a θ_c provede omezení daného signálu.

Konstrukce framu

Zde probíhá, jak už název napovídá, konstrukce použitého framu podle parametrů nastavených v části Nastavení. Vytvoření framu obstarává funkce `frame` z toolboxu LTFAT a daný frame je poté uložen ve struktuře `F`. Pouze v případě, že je zvolen frametype CQT a zapnuto váhování koeficientů, frame se vytvoří funkcí `cqttightframe`. Tato funkce, jejímž autorem je Zdeněk Průša, generuje přímo Parsevalův těsný cqt frame a také reciproké hodnoty vah koeficientů uložené v proměnné `scal`. V případě, že byl frame vytvořen funkcí `frame`, je nutné z něj vytvořit Parsevalův těsný frame, což provede funkce `frametight`. Následně dojde k předpočítání určitých struktur pro konkrétní délku signálu funkcí `frameaccel`, díky kterému bude iterativní volání analýzy a syntézy signálu rychlejší. Na závěr této části ještě program zkontroluje, zda je frame opravdu těsný a zobrazí meze framu, které vrací funkce `framebounds`.

Proximální algoritmus

V této části je volán konkrétní proximální algoritmus. Podle hodnoty nastavené v proměnné `algorithm` se volá buď Condatův algoritmus (`condat.m` – viz část 6.4) nebo Douglas-Rachfordův algoritmus (`doug_rach.m` – viz část 6.5). V případě, že je zapnutá segmentace vstupního signálu, se však nejdříve volá funkce `segmentation` (`segmentation.m` – viz část 6.6), která se stará o rozdělování vstupního signálu na jednotlivé bloky, které pak jednotlivě vstupují do vybraného proximálního algoritmu.

V případě, že byl frame vytvořen funkcí `cqttightframe`, je rekonstruovaný signál v komplexním tvaru a je nutné ho převést zpět do reálného oboru funkcí `real`.

Na konci této části dojde k nahrazení vzorků na spolehlivých pozicích, pokud to bylo v sekci Nastavení povoleno. Program dále spočítá a vypíše ℓ_2 -normu odchylky hodnot na těchto pozicích.

Závěr

V závěru program vykreslí původní a rekonstruovaný signál a pomocí funkce `soundsc` rekonstruovaný signál přehraje. Následně do konzoly vypíše informace o výpočetní době programu. Nakonec pomocí funkce `snr_n` vypočítá SNR hodnoty saturovaného signálu vzhledem k originálnímu, rekonstruovaného signálu vzhledem k originálnímu a jednotlivé hodnoty včetně SNR zlepšení vypíše do konzoly.

6.2 Parametry okna pro DGT

Jak již bylo napsáno v části nastavení, pro DGT lze šířku a posuv okna nastavit ručně, nebo je může spočítat funkce `win_param`. Vstupními parametry této funkce jsou požadovaná délka okna v milisekundách (proměnná `length`), vzorkovací frekvence signálu `fs` a proměnná `rel_shift`, která udává reciprokou hodnotu překryvu vůči velikosti okna. Pokud bude proměnná `rel_shift` nastavená např. na hodnotu 2, okna budou mít překryv 50 %, tedy polovinu velikosti okna.

Funkce podle vstupních parametrů tedy spočítá velikost okna a posun ve vzorcích, avšak tak, aby velikost okna `w` byla dělitelná posunem `a`. Vstupní délka okna v milisekundách je tak pouze orientační a v závislosti na nastaveném posunu se může lišit. Z toho důvodu pak funkce také počítá odchylku vypočítané délky okna od vstupní hodnoty, kterou potom vypíše do konzoly.

6.3 Clipping

K tvrdému omezení vstupního signálu se používá funkce `hard_clip`. Vstupními parametry této funkce jsou signál, dolní úroveň clippingu a horní úroveň clippingu. Vstupní signál lze tedy omezit i asymetricky.

Výstupem této funkce je kromě samotného saturovaného signálu také struktura `masks`, která obsahuje masky signálu definované v části 5.1. Tato struktura tedy obsahuje tři masky pojmenované `Mh`, `Ml` a `Mr`. Masky jsou implementovány jako vektory logických jedniček nebo nul, které jsou stejně dlouhé jako signál.

Součástí balíku pro declipping jsou i dvě funkce pro měkké omezení signálu, neboli soft clipping – `soft_clip.m` a `soft_clip2.m`, které se liší použitou goniometrickou funkcí. Funkce `soft_clip.m` používá funkci arcus tangens, zatímco `soft_clip2.m` provádí clipping podle vztahu (1.2) a používá hyperbolický tangens.

Algoritmus pro declipping je však zatím pouze navržen pro zpracování tvrdě omezeného signálu.

6.4 Condatův algoritmus

Condatův algoritmus v souboru `condat.m` je implementací obecného proximálního algoritmu, upraveného pro potřeby declippingu (viz část 5.3.1). Do funkce vstupuje satureovaný signál a struktura `param`, ze které si algoritmus načte všechny potřebné proměnné, např. frame `F`, masky signálu, úroveň clippingu `theta`, aj.

V první části kódu je sekce nastavení, kde je možné nastavit několik parametrů. Všechny parametry tohoto algoritmu jsou součástí struktury `paramsolver`. Prvním z nich je parametr `verbose`, který, pokud je nenulový, umožní programu vypisovat do konzoly aktuální číslo iterace a relativní změnu v koeficientech oproti minulé iteraci. Parametr `maxit` je ukončovací parametr, který nastavuje maximální počet iterací algoritmu. Kromě něho je použit i ukončovací parametr `coef_tol`, který představuje nejmenší relativní změnu koeficientů. Pokud bude tato změna v aktuální iteraci menší, než je nastaveno v `coef_tol`, algoritmus se ukončí. Funkce je naprogramována tak, že po skončení algoritmu vypíše, kolik iterací proběhlo a z jakého důvodu došlo k ukončení. Další tři možné parametry jsou `tau`, `sigma` a `ro`, což jsou přímo koeficienty proximálního algoritmu. Jejich funkce je popsána v části 5.3.1.

Pokud je zapnuto váhování koeficientů, z proměnných `tau`, `scal` a `weightCoef` jsou spočítány prahy pro měkké prahování vzhledem k jednotlivým koeficientům a následně uloženy do proměnné `T`, která je v tomto případě vektor. Pokud je váhování vypnuto, proměnná `T` je skalár a je rovna hodnotě `tau`.

Jako startovací bod algoritmu jsou použity koeficienty satureovaného signálu, ale je možné také použít nulový vektor. Volba startovacího bodu však nemá na výsledný rekonstruovaný signál velký vliv. Bylo totiž vyzkoušeno, že v prvních několika iteracích dochází k výrazným změnám koeficientů nezávisle na volbě startovacího bodu. Postupně pak algoritmus konverguje k řešení.

Lineární transformace, které v algoritmu probíhají, tedy analýza a syntéza signálu, jsou počítány pomocí funkcí `frana` a `frsyn` z toolboxu LTFAT. V průběhu iterací je kromě samotných koeficientů signálu počítána také ℓ_1 -norma koeficientů, která je ukládána do vektorové proměnné `l1_norm`. Tato proměnná je pak také nastavena jako výstupní proměnná funkce, je tedy možné po skončení algoritmu porovnat průběh ℓ_1 -normy koeficientů.

Vlastní implementace algoritmu vychází z jeho popisu v části 5.3.1, proto není nezbytné ji zde podrobněji popisovat.

6.5 Douglas-Rachfordův algoritmus

Podobně jako v případě Condatova algoritmu, je i Douglas-Rachfordův algoritmus v souboru `doug_rach.m` implementací algoritmu přestaveného v teoretické části 5.4.2. Douglas-Rachfordův algoritmus je zde implementován obecněji než Condatův. Vstupními parametry jsou, kromě saturovaného signálu a struktury `param`, také funkce `f1` a `f2`, které jsou pomocí anonymních funkcí definovány těsně před voláním funkce `doug_rach` v hlavním souboru. Každá funkce je vlastně struktura, která v proměnné `eval` obsahuje svoji definici a v proměnné `prox` pak definuje proximální operátor dané funkce.

Struktura funkce `doug_rach` je velmi podobná struktuře Condatova algoritmu. Na prvním místě je opět sekce nastavení, kde lze nastavovat parametr `verbose`, maximální počet iterací `maxit` a ukončovací parametr (nyní jak pro relativní změnu koeficientů v parametru `coef_tol`, tak i pro relativní změnu objektivní funkce `tol`). Dále je možné nastavit velikost kroku `lambda` a konvergenční parametr `gamma`.

Stejně jako v případě Condatova algoritmu, pokud není splněna alespoň jedna podmínka, algoritmus se ukončí a vypíše, kolik iterací proběhlo a důvod, proč se algoritmus ukončil.

Kromě rekonstruovaného signálu pak funkce vrací také vektor hodnot objektivní funkce v průběhu iterací.

6.6 Segmentace

Funkce `segmentace` zpracovává vstupní signál, pokud je v nastavení segmentace zapnutá. Vstupními parametry jsou saturovaný signál a struktura `param`. Podle nastavených hodnot délky zpracovávaného segmentu v proměnné `seg` a velikosti přesahu v proměnné `overlap` funkce vybírá jednotlivé segmenty signálu a následně pro ně volá proximální algoritmus. Rekonstruované bloky jsou následně zpět seskládány do jednoho zvukového signálu. Místa s překryvem se následně vypočítají jako aritmetický průměr hodnot z obou bloků. Tato funkce se tedy chová jako mezivrstva mezi hlavním souborem declippingu a proximálním algoritmem.

6.7 Projekce

Zatímco u Condatova algoritmu je projekce na přípustnou množinu řešena přímo v kódu souboru `condat.m`, v případě Douglas-Rachfordova algoritmu je tato projekce implementována v souboru `proj_parse_frame.m`, který je volán z `f2.prox`, jakožto proximální operátor indikátorové funkce.

Tato projekce je teoreticky popsána v části 5.4.3, konkrétně ve vztahu (5.30). Vstupními proměnnými této funkce je vektor koeficientů \mathbf{c} , struktura s parametry `param` a saturovaný signál. Výstupem je pak samotný vektor projektovaných koeficientů `proj`.

Funkce nejprve spočítá syntézu vstupního signálu (v teoretické části značeno jako \mathbf{Az}). Dále stanoví vektory \mathbf{b}_1 a \mathbf{b}_2 podle (5.25). Samotnou projekci podle vztahu (5.31) řeší funkce `proj_box` (autorem je Pavel Rajmic), která je také dostupná v toolboxu UNLocBoX.

7 EXPERIMENTY A VÝSLEDKY

V této kapitole jsou prezentovány výsledky algoritmu pro declipping popsaného v předchozích kapitolách.

Nejprve jsou v části 7.1 uvedeny základní informace o testovaných signálech. V části 7.2 je provedeno srovnání implementace declippingu pomocí Condatova algoritmu a Douglas-Rachfordova algoritmu. Z důvodů uvedených v této části bude v testech použita pouze varianta declippingu s využitím Douglas-Rachfordova algoritmu.

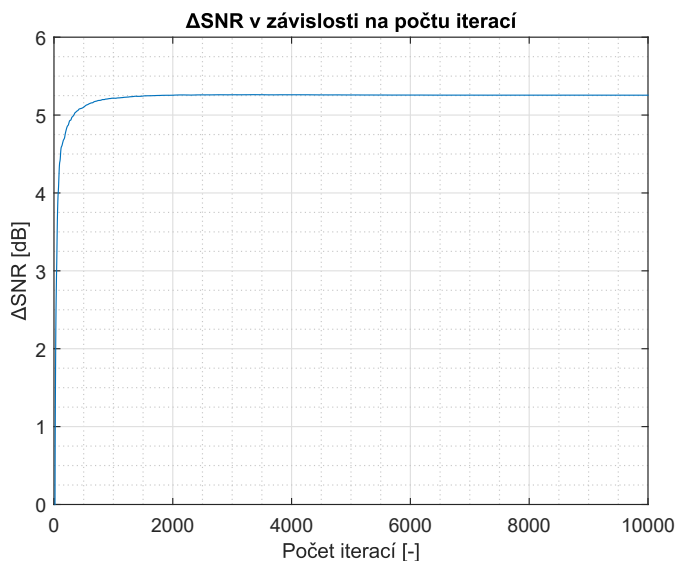
V dalších dvou částech jsou shrnuty výsledky testovaných algoritmů pomocí ukazatele ΔSNR (část 7.3), resp. PEMO-Q (část 7.4). V obou částech jsou demonstrovány výsledky pro verze algoritmů používající DGT a různé modifikace CQT. V závěru těchto částí je uvedeno srovnání těchto metod s metodou ClipFix, která je implementovaná v audio editoru Audacity, a algoritmem A-SPADE, což je v dnešní době pravděpodobně nejkvalitnější algoritmus pro rekonstrukci saturovaných signálů.

Dále následuje část porovnávající použité algoritmy z hlediska výpočetní náročnosti. Závěr této kapitoly pak tvoří výsledky subjektivního srovnání jednotlivých metod pomocí poslechových testů.

7.1 Testování

Testování jednotlivých algoritmů probíhá na pěti zvukových signálech – akustická kytara, kontrabas, znělka, řečový signál a orchestr. Použité zvukové signály mají vzorkovací frekvenci 16 kHz a jsou ve formátu *.wav.

Každý zvukový signál byl postupně omezen na úroveň $\theta_c \in \{0,1; 0,2; \dots; 0,9\}$ a tyto signály byly následně rekonstruovány testovanými algoritmy.



Obr. 7.1: Typický průběh ukazatele ΔSNR v závislosti na počtu iterací.

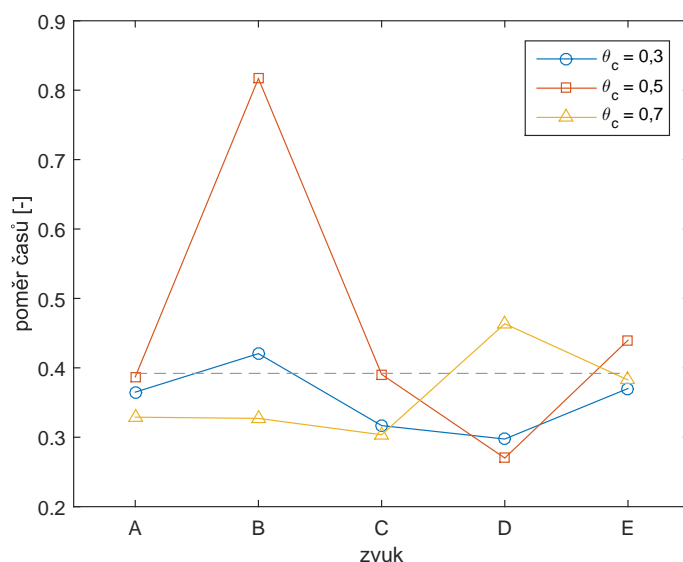
Pro rekonstrukční algoritmus bylo jako ukončovací kritérium pevně stanoveno 2000 iterací. Z proměřené závislosti ΔSNR na počtu iterací je patrné, že při vyšším počtu iterací se hodnota ΔSNR prakticky nemění, což znamená, že dalšími iteracemi se již výsledek nijak výrazně nevylepší. Tuto závislost lze pozorovat na grafu 7.1.

7.2 Srovnání proximálních algoritmů

V této části je provedeno stručné srovnání Condatova algoritmu (viz část 5.3.1) a Douglas-Rachfordova algoritmu (viz část 5.4.2).

Podle teoretických předpokladů by oba proximální algoritmy měly konvergovat ke stejnému výsledku. Nevýhodou Condatova algoritmu je v případě declippingu především nutnost počítání tří projekcí (na množiny R , H a L). To s sebou nese jednak nepřesnosti v rekonstrukci (výsledná projekce v jedné iteraci se počítá jako součet jednotlivých projekcí, proto např. vzorky na spolehlivých pozicích nejsou totožné s originálním signálem, ale mírně se odchyľují), tak i v časové náročnosti, kde v každé iteraci musíme počítat tři analýzy a dvě syntézy, což jsou časově nejnáročnější operace algoritmu.

Algoritmus Douglas-Rachford naopak díky projekci popsané v části 5.4.3 provede pouze jednu analýzu a jednu syntézu. Navíc všechny vzorky na spolehlivých pozicích přesně odpovídají původnímu signálu.



Obr. 7.2: Poměr výpočetních časů pro Condatův algoritmus a Douglas-Rachfordův algoritmus v případě použití DGT.

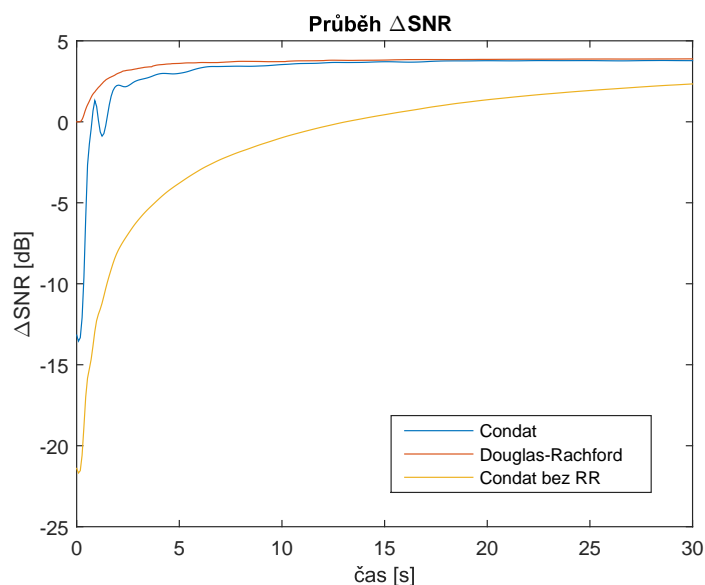
Na grafu 7.2 můžeme vidět porovnání výpočetních časů Condatova a Douglas-Rachfordova algoritmu v případě použití DGT. Porovnání proběhlo pro pět zvuko-

vých ukázek a tři úrovně zkreslení ($\theta_c = \{0,3; 0,5; 0,7\}$). Oba algoritmy byly ukončeny ve chvíli, kdy součet absolutních hodnot diferencí hodnot SNR z posledních deseti iterací byl menší než 0,0001. Předpokládá se tedy, že pokud se hodnota SNR výrazně nemění, algoritmus dokonvergoval k řešení.

Z grafu je patrné, že až na jeden případ je pro testované signály Douglas-Rachfordův algoritmus minimálně dvakrát rychlejší než Condatův algoritmus. Šedou čárkovanou čarou je v grafu vyznačen průměr poměru časů, který je v tomto případě 0,392.

Na grafu 7.3 lze vidět srovnání průměrného průběhu hodnoty ΔSNR pro Condatův algoritmus a Douglas-Rachfordův algoritmus v závislosti na čase. Definice výpočtu hodnoty SNR i ΔSNR je uvedena v části 7.3. Ze srovnání křivek v tomto grafu lze vyčíst, že Douglas-Rachfordův algoritmus konverguje rychleji a stabilněji než Condatův. Průběhy ΔSNR byly měřeny pouze na saturovaných vzorcích.

Pro zajímavost je zde žlutou barvou uveden i průběh ΔSNR pro Condatův algoritmus bez nahrazování spolehlivých vzorků (viz kapitola 6, RR = *replace reliable*). Hodnota ΔSNR se pak počítá z celého signálu, nejen ze saturovaných vzorků. V tomto případě je vidět výrazný pokles v kvalitě rekonstrukce, způsobený odchylkou vzorků na spolehlivých pozicích vlivem nepřesné projekce.



Obr. 7.3: Porovnání průměrných průběhů ΔSNR pro Condatův algoritmus a Douglas-Rachfordův algoritmus.

Z uvedených výsledků lze odvozovat, že Douglas-Rachfordův algoritmus s největší pravděpodobností konverguje ke stejnému řešení, jako Condatův algoritmus, avšak mnohem rychleji. Proto v dalších částech práce bude pro výpočty použita pouze verze s Douglas-Rachfordovým algoritmem.

7.3 Porovnání pomocí ukazatele SNR

Jak již bylo popsáno v úvodu kapitoly, testování kvality rekonstrukčních algoritmů je provedeno několika metodami. V této části budou výsledky jednotlivých metod srovnány pomocí ukazatele SNR, resp. ΔSNR .

Ukazatel SNR je definován vztahem:

$$\text{SNR}(\mathbf{u}, \mathbf{v}) = 20 \log_{10} \left(\frac{\|\mathbf{u}\|_2}{\|\mathbf{u} - \mathbf{v}\|_2} \right), \quad (7.1)$$

kde \mathbf{u} představuje původní signál a \mathbf{v} rekonstruovaný signál. Jako hlavní kritérium hodnocení je však v této práci použit ukazatel ΔSNR , což je rozdíl SNR rekonstruovaného a saturovaného signálu. Za předpokladu, že tyto signály se na spolehlivých pozicích rovnají, vyjadřuje ΔSNR zlepšení SNR oproti saturovanému signálu na saturovaných pozicích signálu. ΔSNR je vypočten jako

$$\Delta\text{SNR} = \text{SNR}(\mathbf{y}, \hat{\mathbf{y}}) - \text{SNR}(\mathbf{y}, \mathbf{y}_c), \quad (7.2)$$

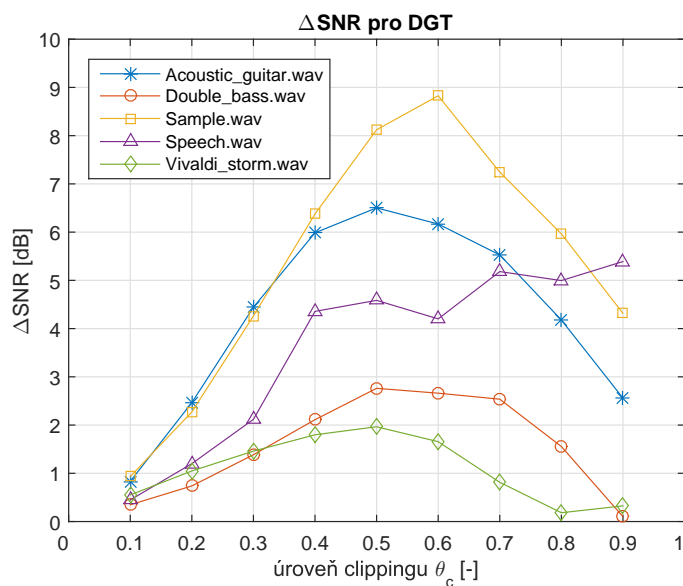
kde \mathbf{y} značí původní signál, \mathbf{y}_c značí saturovaný signál a $\hat{\mathbf{y}}$ pak rekonstruovaný signál. Díky tomuto ukazateli lze určit, o kolik dB metoda vylepšila saturovaný signál.

V následujících podčástech jsou zobrazeny výsledné hodnoty ΔSNR pro použité typy framů DGT, CQT a její modifikace. V závěru této části jsou tyto metody srovnány s konkurenčními metodami.

7.3.1 DGT

Parametry diskrétní Gaborovy transformace byly pro testování nastaveny podobně, jako v případě konkurenčních metod uvedených v části 1.3. To znamená, že bylo použito Hammingovo okno o délce $N = 1024$ vzorků, což při vzorkovací frekvenci zvukových signálů 16 kHz dává délku okna 64 ms a překryv sousedních oken byl 75 %. Pro každé okno bylo použito 2048 kmitočtových kanálů. Při těchto parametrech tvoří DGT těsný frame.

Na grafu 7.4 je znázorněn ukazatel ΔSNR pro pět testovaných zvukových signálů v závislosti na úrovni clippingu θ_c . Je možné zde pozorovat rozdílné výsledky pro různé zvukové signály. Tyto rozdílné výsledky jsou dány především řídkostí zvukového signálu v použitém slovníku. Čím řidší signál, tím přesnější je aproximace řídkým modelem a tím pádem i kvalitnější rekonstrukce, tedy i vyšší hodnota ΔSNR .



Obr. 7.4: Hodnoty ukazatele ΔSNR pro signály se slovníkem DGT.

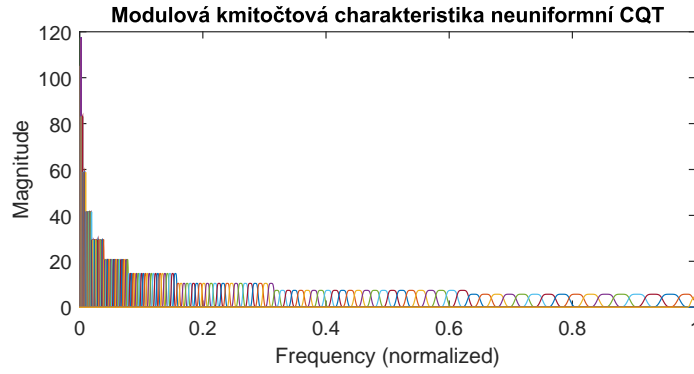
7.3.2 CQT banka filtrů

V případě CQT banky filtrů je nastavení jiné, než v případě DGT. Je zde potřeba nastavit minimální a maximální frekvenci banky filtrů a dále také počet filtrů na oktávu. Pro všechny případy byla použita minimální frekvence 10 Hz, maximální frekvence 8 kHz (polovina vzorkovacího kmitočtu) a 24 filtrů na oktávu. V následujících podčástech jsou prezentovány výsledky podle ukazatele ΔSNR pro různé typy CQT.

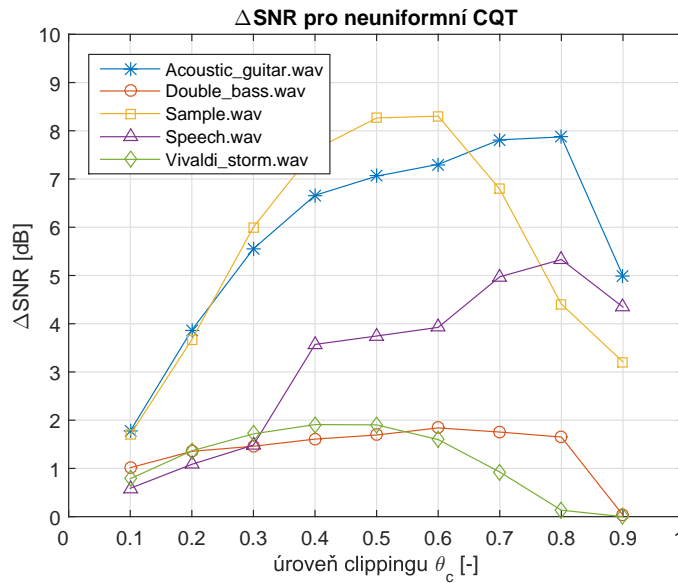
Neuniformní CQT

Jako první případ je zde uvedena neuniformní banka filtrů CQT, která má v rámci oktávy konstantní koeficient podvzorkování. Tento typ banky filtrů je v toolboxu LTFAT označován jako `'regsampling'` a je to výchozí typ. Modulová kmitočtová charakteristika použité banky filtrů je zobrazena v grafu 7.5.

Na následujícím grafu 7.6 jsou vykresleny hodnoty ukazatele ΔSNR pro pět testovaných zvukových signálů, obdobně jako tomu bylo v případě DGT. Ze srovnání hodnot ΔSNR pro slovníky DGT a CQT lze vyčíst, že CQT funguje lépe pro vysoké míry zkreslení satureovaného signálu ($\theta_c \leq 0,3$). Lepších výsledků dosahuje CQT hlavně pro signál akustické kytary. Naopak řečový signál a signál kontrabasu byl lépe zpracovaný pomocí slovníku DGT.



Obr. 7.5: Modulová kmitočtová charakteristika neuniformní banky filtrů CQT.



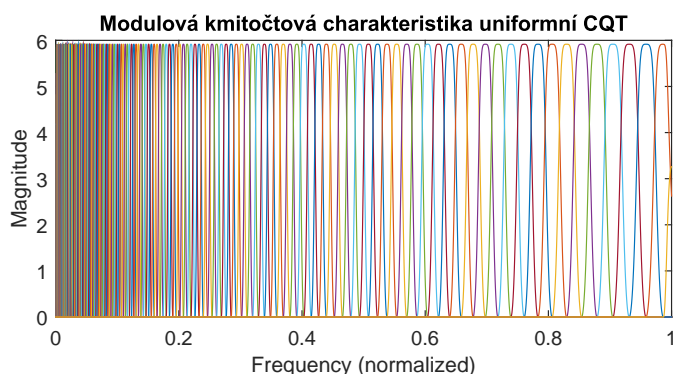
Obr. 7.6: Hodnoty ukazatele ΔSNR pro signály s použitím neuniformní CQT.

Uniformní CQT s váhováním koeficientů

V dalším případě byla testována uniformní banka filtrů CQT. Protože u rovnoměrné banky filtrů je energie obsažená v impulsních charakteristikách filtrů s vyššími frekvencemi vyšší (viz část 4.5), tato možnost při declippingu preferuje kmitočty s vyšší frekvencí. Tento jev však není pro rekonstrukci saturovaného signálu žádoucí a výsledná kvalita rekonstrukce nedosahuje takových výsledků jako v případě neuniformní banky filtrů nebo DGT. Z tohoto důvodu zde výsledky pro uniformní CQT nejsou uvedeny.

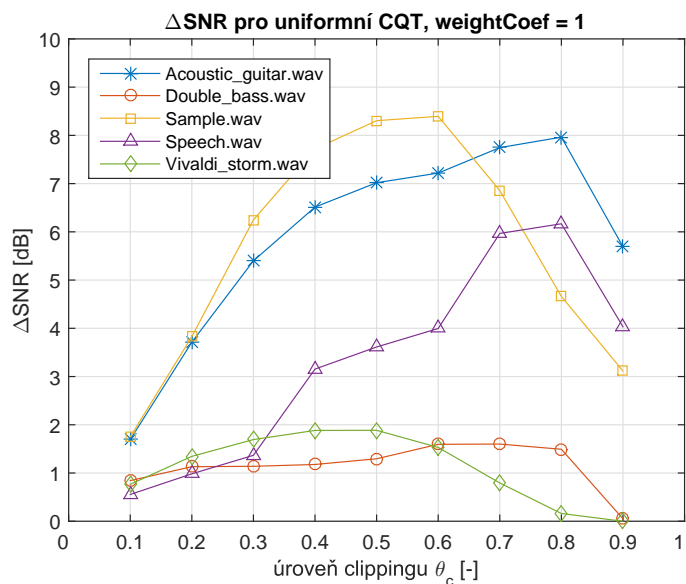
Avšak s použitím váhování koeficientů (viz poslední odstavec části 4.5) jsme schopni tento nedostatek uniformní banky filtrů odstranit. V tomto případě jsou koeficienty váhovány tak, aby se banka filtrů chovala tak, že ℓ_2 -normy modulových kmitočtových charakteristik jsou konstantní a exponent `weightCoef` = 1.

Na grafu 7.7 je zobrazena modulová kmitočtová charakteristika použité uniformní banky filtrů. Tato banka filtrů má koeficient podvzorkování konstantní pro všechny filtry a jednotlivé filtry jsou normalizovány „na špičku“ (*peak normalized*).



Obr. 7.7: Modulová kmitočtová charakteristika uniformní banky filtrů CQT.

Graf 7.8 vykresluje hodnoty ΔSNR pro CQT s váhováním koeficientů. Na první pohled se hodnoty nijak výrazně neodlišují od verze s neuniformní CQT. Malé zlepšení lze zaznamenat pro řečový signál s úrovněmi clippingu $\theta_c = 0,7$ a $0,8$. Naopak u signálu kontrabasu došlo podle ΔSNR k mírnému zhoršení rekonstrukce. Uvedené rozdíly jsou však minimální a poslechově prakticky nerozeznatelné.

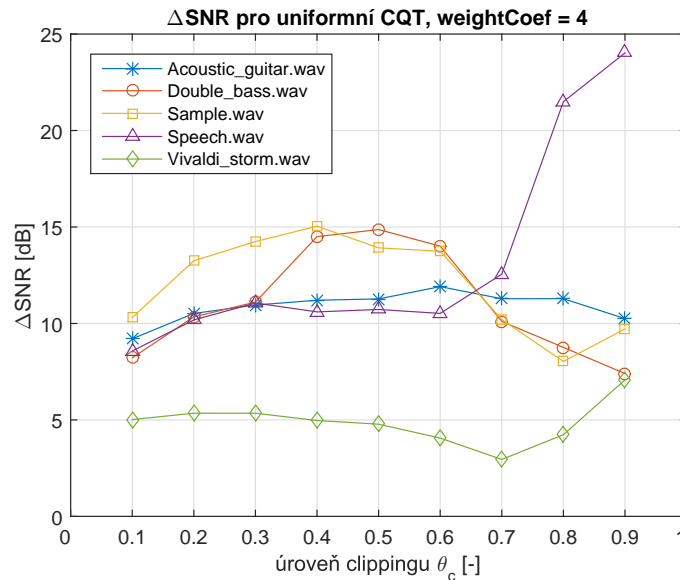


Obr. 7.8: Hodnoty ukazatele ΔSNR pro signály s použitím uniformní CQT a váhováním koeficientů.

Uniformní CQT s váhováním koeficientů a potlačením vysokofrekvenčních složek

Poslední testovanou modifikací banky filtrů CQT je také uniformní banka filtrů s váhováním koeficientů. Modulová kmitočtová charakteristika je v tomto případě stejná jako v předchozím případě (graf 7.7). V tomto případě však navíc dochází k potlačení vysokofrekvenčních složek – při rekonstrukci jsou tedy preferovány nižší kmitočty. Toho je dosaženo empirickým exponentem `weightCoef`, pro který musí platit, že `weightCoef` > 1. Po sérii testů bylo zjištěno, že nejlepší kvalita rekonstrukce podle SNR je pro `weightCoef` = 4.

Na grafu 7.9 jsou vykresleny hodnoty ΔSNR pro použitou uniformní banku filtrů s váhováním koeficientů a potlačením vysokofrekvenčních složek pro `weightCoef` = 4. Již na první pohled si lze všimnout velkého nárůstu hodnot ΔSNR , v průměru o 7 dB, oproti předchozí variantě bez potlačení vysokofrekvenčních složek.



Obr. 7.9: Hodnoty ukazatele ΔSNR pro signály s použitím uniformní CQT, váhováním koeficientů a potlačením vysokofrekvenčních složek.

Největší přínos potlačení vyšších frekvencí se pro testované signály projevil u nahrávky kontrabasů a řečového signálu (průměrný nárůst ΔSNR o 10 dB oproti předchozí variantě). U řečového signálu se tento nárůst projevil především pro nepatrné hodnoty zkreslení signálu $\theta_c \geq 0,8$. Nejméně se projeвило potlačení vysokých frekvencí ΔSNR pro signál orchestru, kde průměrný nárůst ΔSNR je 3,7 dB.

7.3.3 Srovnání s konkurenčními metodami

V této části jsou představené algoritmy porovnány nejen mezi sebou, ale i se dvěma dalšími metodami.

Z představených metod zde bude porovnána varianta s použitím DGT, neuniformní CQT (CQT_nonuniform) a také uniformní CQT s váhováním a exponentem `weightCoef = 4` (označovaná jako CQT_rew4).

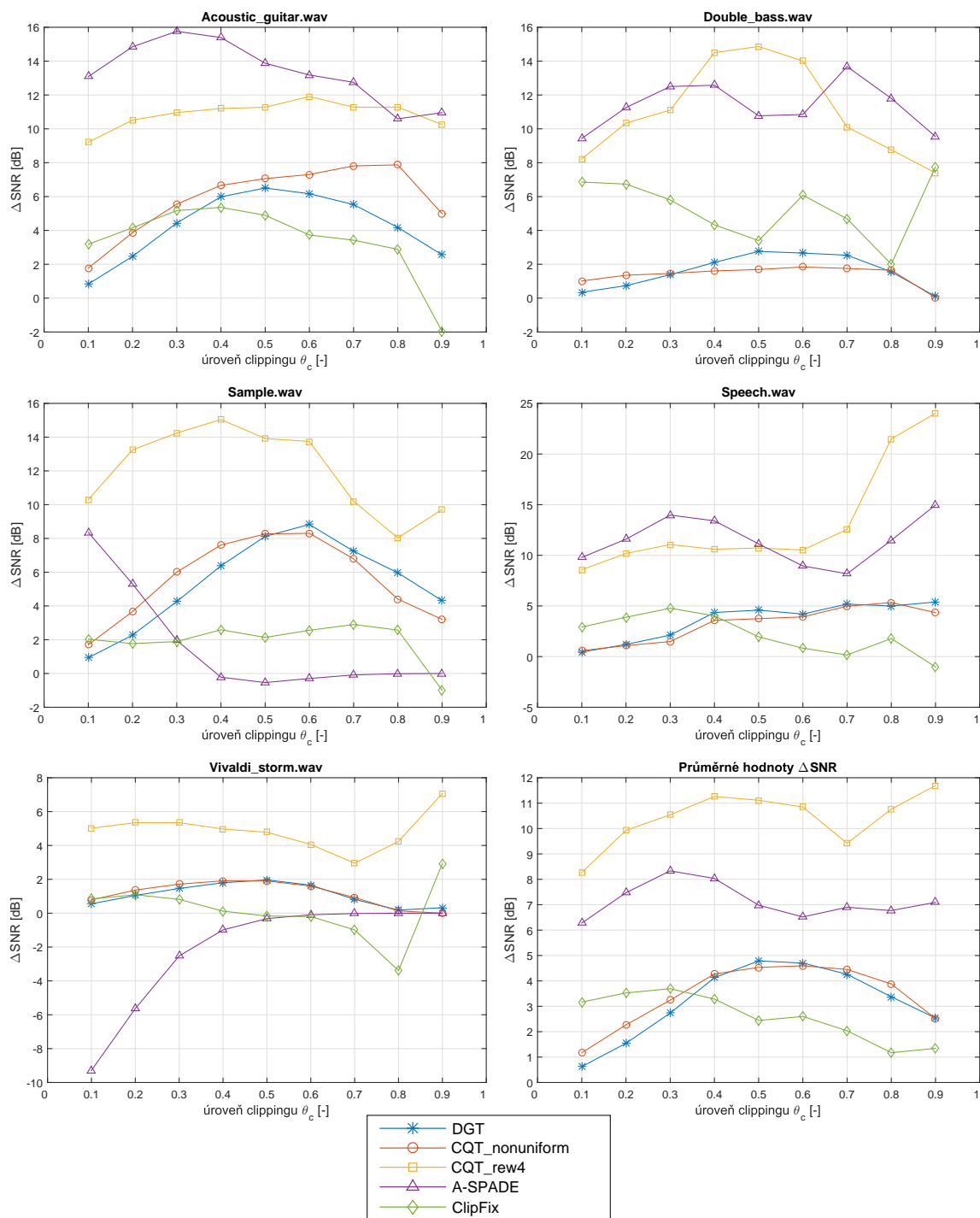
První z konkurenčních metod, se kterou jsou tyto varianty porovnány, je algoritmus ClipFix [43], který je implementován v audio editoru Audacity a funguje na principu kubické interpolace saturovaných vzorků. Výhodou této metody je především rychlost zpracování, avšak autoři uvádějí, že se hodí pouze pro lehké zkreslení vstupního signálu.

Další porovnávanou metodou je algoritmus A-SPADE, který byl detailněji představen v závěru kapitoly 2. Implementace algoritmu umožňuje zvolit si mezi dvěma přednastavenými profily rekonstrukce, a to profilem „accuracy“, který je nastaven pro maximalizaci kvality rekonstrukce bez ohledu na čas výpočtu, a profilem „fast“, který představuje kompromis mezi kvalitou rekonstrukce a výpočetním časem. Tyto profily se liší především redundancí použité báze, velikostí překryvu zpracovaných bloků a parametrem představujícím cílovou řídkost rekonstruovaného signálu. Pro účely tohoto srovnání byly použity výstupy algoritmu A-SPADE s profilem „accuracy“. Výsledky algoritmu by tedy měly být nejlepší možné.

Na grafech 7.10 lze pozorovat hodnoty ΔSNR pro testované metody v závislosti na úrovni clippingu θ_c . Každý graf představuje výsledky naměřené pro konkrétní testovaný audio signál. Poslední graf (vpravo dole) vykresluje průměrné hodnoty ΔSNR , získané z předchozích výsledků.

Z grafů je patrné, že výsledky metod s použitím DGT i neuniformní CQT jsou velmi podobné pro všechny testované audio signály. Při jejich srovnání s metodou ClipFix lze říci, že ClipFix poskytuje podle ΔSNR lepší výsledky pro silně zkreslené signály ($\theta_c \geq 0,3$) než představená metoda s využitím DGT nebo neuniformní CQT, avšak pro méně zkreslené signály je tomu naopak. Velmi dobré hodnoty ΔSNR podávají metody A-SPADE a CQT_rew4. Pro audio signál Vivaldi_storm.wav však metoda A-SPADE nevylepšila SNR oproti saturovanému signálu, naopak jej ještě zhoršila ($\Delta\text{SNR} < 0$). U signálu Sample.wav zvýšila hodnotu SNR pouze v případě silného zkreslení.

Z celkového srovnání vyplývá, že nejlepší rekonstrukci z hlediska ΔSNR poskytuje metoda CQT_rew4, která podává vysoké hodnoty ΔSNR nezávisle na testovaném audiosignálu.



Obr. 7.10: Srovnání testovaných algoritmů podle hodnot ΔSNR .

7.4 Porovnání pomocí PEMO-Q

Obdobně jako v předchozí části 7.3, i v této části budou nejprve vyhodnoceny výsledky představené metody pomocí Douglas-Rachfordova algoritmu s použitím DGT a následně různých modifikací CQT. V závěru budou tyto výsledky srovnány se dvěma konkurenčními metodami A-SPADE a ClipFix.

V této části však bude srovnání provedeno pomocí PEMO-Q [44, 45], což je softwarový nástroj pro objektivní hodnocení kvality zvukových signálů. Na rozdíl od SNR však do výpočtu aplikuje počítačový model lidského slyšení, díky čemuž by výsledky měly lépe odpovídat lidskému slyšení.

Současná implementace PEMO-Q je v prostředí MATLAB. K vyhodnocení zvukové kvality se používá funkce `audioqual` v případě hudebního signálu a `speechqual` v případě řečového signálu. Výstupem těchto funkcí jsou hodnoty PSM (*Perceptual Similarity Measure*) udávající celkovou korelaci interních reprezentací, PSM_t udávající pátý kvantil vážených hodnot $PSM(t)$, ODG (*Objective Difference Grade*) udávající hodnotu PSM_t transformovanou na škálu ODG od -4 do 0 , a vektor `PSM_inst`, který obsahuje jednotlivé hodnoty $PSM(t)$ v časech t .

Přepočet na škálu ODG je v tomto algoritmu přizpůsoben poslechovým testům, proto funguje pouze pro výchozí hodnoty parametrů. Jedním z těchto parametrů je i vzorkovací frekvence, jejíž výchozí hodnota je $44,1$ kHz. Testované signály v této práci však mají vzorkovací frekvenci 16 kHz. PEMO-Q tedy není schopen převést hodnoty PSM_t na ODG, a proto budou uvedené výsledky porovnány pomocí PSM_t .

Následující podčásti shrnou výsledky pomocí PEMO-Q obdobně, jako tomu bylo v předchozí části. Nejprve budou zobrazeny výsledné hodnoty pro použité typy framů DGT, CQT a její modifikace a v závěru lze najít opět srovnání s konkurenčními metodami.

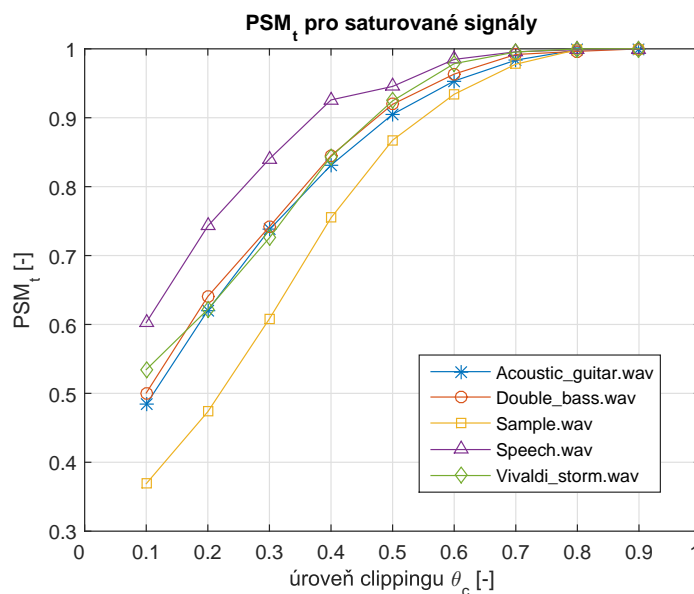
7.4.1 Saturované signály

Ukazatel PSM_t udává, na kolik je zkreslení v signálu patrné, avšak neříká nic o tom, o kolik metoda vylepšila původní saturovaný signál. Proto jsou v této části uvedeny výsledky PSM_t pro nezpracované saturované signály. Pomocí nich bude možné následně lépe porovnat, o kolik jednotlivé metody z hlediska PSM_t vylepšily testovaný signál.

Na grafu 7.11 lze vidět hodnoty ukazatele PSM_t pro pět testovaných zvukových ukázek v závislosti na úrovni clippingu θ_c . Z grafu je patrné, že pro stejné hodnoty úrovně clippingu θ_c mají signály různé hodnoty PSM_t . To souvisí především s faktem, že i když byly signály před clippingem normalizovány, u každého signálu byl ořezán jiný počet vzorků.

Z grafu lze dále vyčíst, že pro nízké úrovně zkreslení signálu ($\theta_c \geq 0,8$) jsou hodnoty PSM_t téměř rovny jedné. Z poslechového hlediska by tedy podle hodnot PSM_t měly být tyto signály téměř nerozeznatelné od originálního nezkresleného signálu.

Pro více zkreslené zvukové signály lze při hodnocení podle PSM_t tvrdit, že artefakty způsobené zkreslením signálu budou nejméně slyšitelné v případě řečového signálu (vyšší hodnoty ukazatele PSM_t). Naopak v případě signálu *Sample.wav* a *Vivaldi_storm.wav* by zkreslení mělo být poslechově největší.



Obr. 7.11: Hodnoty ukazatele PSM_t pro saturované signály.

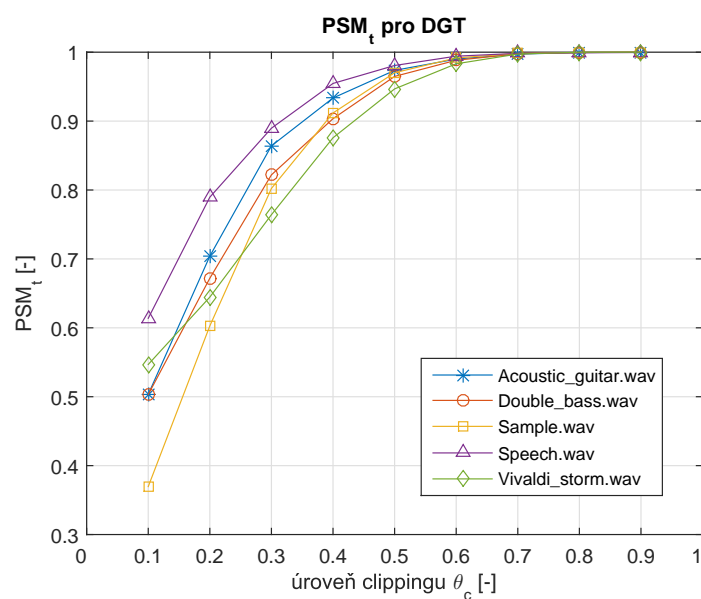
7.4.2 DGT

V této části jsou prezentovány výsledky představené metody s použitím slovníkem DGT. Nastavení bylo identické, jako v předchozím případě (viz část 7.3.1).

Graf 7.12 zobrazuje hodnoty ukazatele PSM_t pro testované zvukové signály zpracované s použitím DGT slovníku v závislosti na úrovni clippingu θ_c .

Z grafu vyplývá, že pro velké zkreslení ($\theta_c = 0,1$) nedošlo aplikováním této metody k výraznému zlepšení saturovaného signálu. K nepatrnému nárůstu kvality došlo podle ukazatele PSM_t pouze u řečového signálu a signálu akustické kytary.

Pro méně zkreslené signály pak pro testované signály dochází ke znatelnému zlepšení hodnot PSM_t . Nejvíce je zlepšení patrné pro zvukový signál *Sample.wav*. Naopak nejméně se zlepšení projevilo u signálů *Double_base.wav* a *Vivaldi_storm.wav*, tedy u signálů s nejmenší řídkostí.

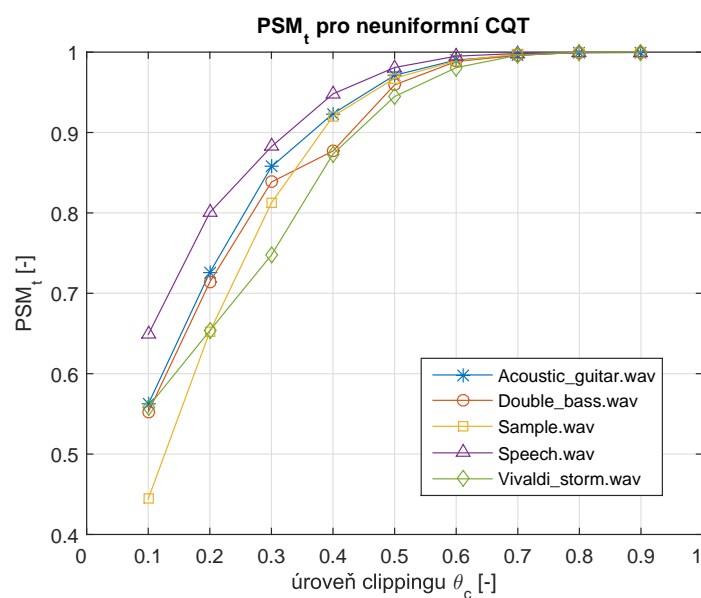


Obr. 7.12: Hodnoty ukazatele PSM_t pro signály se slovníkem DGT.

7.4.3 CQT banka filtrů

Neuniformní CQT

Graf 7.13 vykresluje hodnoty PSM_t pro neuniformní CQT banku filtrů. Při srovnání hodnot PSM_t za použití DGT a neuniformní CQT se potvrzuje stejný předpoklad, jako při porovnání hodnot ΔSNR , a sice ten, že CQT funguje lépe pro vysoké úrovně zkreslení, naopak pro lehké zkreslení podává mírně kvalitnější výsledky DGT.



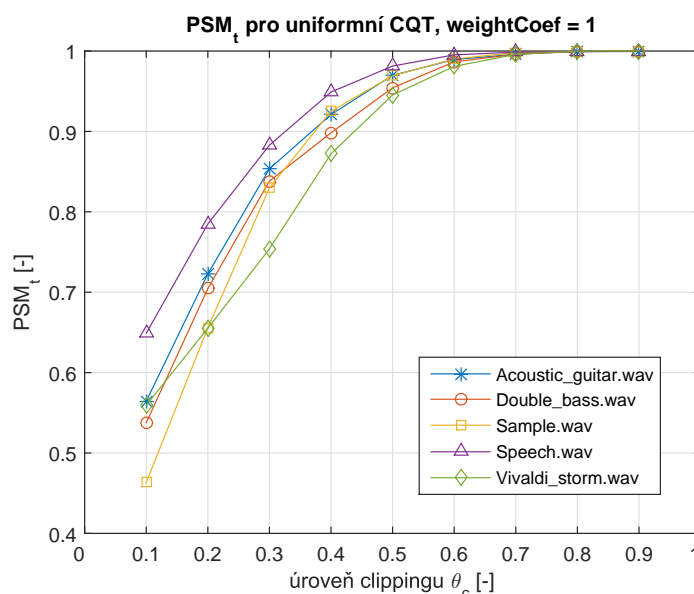
Obr. 7.13: Hodnoty ukazatele PSM_t pro signály s použitím neuniformní CQT.

Uniformní CQT s váhováním koeficientů

Jak již bylo konstatováno při srovnávání hodnot ΔSNR , rozdíly v kvalitě rekonstrukce mezi neuniformní CQT a uniformní CQT s váhováním koeficientů jsou téměř nulové. Totéž platí i pro srovnání hodnot PSM_t .

Nepatrné zlepšení je možné pozorovat v případech signálu kontrabasu pro úroveň clippingu $\theta_c = 0,4$ nebo u signálu Sample pro úrovně clippingu $\theta_c = 0,1$ a $0,3$. Naopak mírné zhoršení je patrné např. pro řečový signál při zkreslení $\theta_c = 0,2$.

Hodnoty PSM_t pro tento případ CQT jsou vykresleny v grafu 7.14.



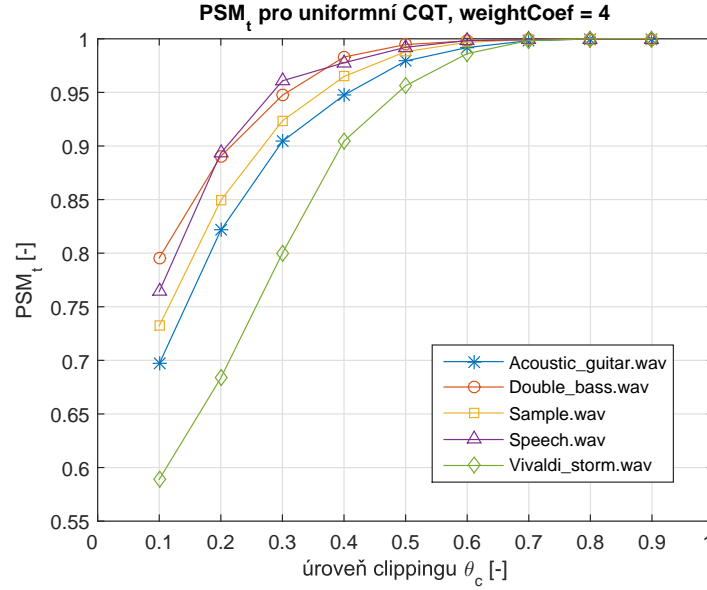
Obr. 7.14: Hodnoty ukazatele PSM_t pro signály s použitím uniformní CQT a váhováním koeficientů.

Uniformní CQT s váhováním koeficientů a potlačením vysokofrekvenčních složek

Potlačení vysokofrekvenčních složek při porovnání podle ΔSNR způsobilo znatelný nárůst dB, avšak pro některé zvukové signály se toto potlačení poslechově negativně projeví úbytkem vysokých frekvencí v celém signálu. Cílem však není vyvinout metodu, která bude podávat nejlepší hodnoty SNR, ale jejíž výsledky budou perceptuálně nejkvalitnější. I z tohoto důvodu je srovnání ukazatelů PSM_t velice zajímavé.

Hodnoty PSM_t pro uniformní CQT banku filtrů s váhováním koeficientů a potlačením vysokofrekvenčních složek jsou vykresleny v grafu 7.15. Už na první pohled si lze, podobně jako v případě ΔSNR , všimnout nárůstu hodnot PSM_t oproti předchozím variantám algoritmu. Nejvýraznější zlepšení lze opět zaznamenat u signálu

kontrabasů a znělky Sample.wav. K nejmenší změně došlo v případě signálu orchestru, avšak i zde lze zaznamenat nepatrné zvýšení hodnot ukazatele PSM_t .



Obr. 7.15: Hodnoty ukazatele PSM_t pro signály s použitím uniformní CQT, váhováním koeficientů a potlačením vysokofrekvenčních složek.

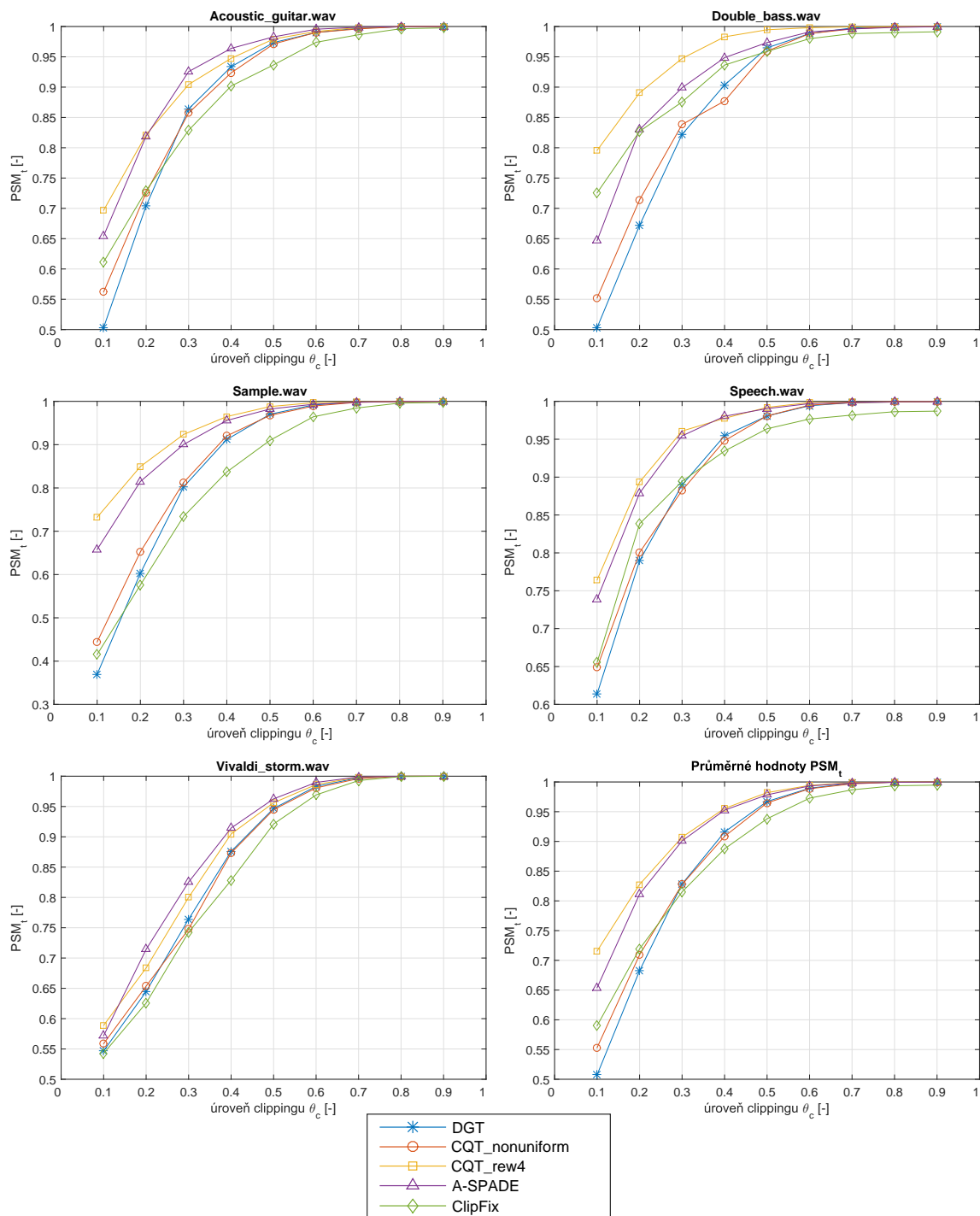
7.4.4 Srovnání s konkurenčními metodami

V této části je opět provedeno srovnání představeného algoritmu pro declipping a jeho modifikací s metodami A-SPADE a ClipFix, nyní však pomocí ukazatele PSM_t . Na grafech 7.16 je vynesena závislost hodnot PSM_t na úrovni clippingu θ_c . Každý graf opět představuje výsledky pro konkrétní zvukový signál. Výsledky z těchto pěti grafů jsou následně zprůměrovány a vykresleny do posledního grafu (vpravo dole).

Z uvedených grafů vyplývá, že výsledky metod s použitím DGT i neuniformní CQT jsou opět velmi podobné pro všechny testované audio signály. Nejvíce se liší pro nejvýraznější zkreslení ($\theta_c = 0,1$), kde je neuniformní CQT lepší. Naopak pro $\theta_c \geq 0,3$ podává nepatrně lepší výsledky DGT.

V případě metody ClipFix můžeme tvrdit, že tato metoda podává kvalitnější výsledky než CQT a DGT pro silně zkreslené signály ($\theta_c \leq 0,2$), avšak pro $\theta_c \geq 0,3$ je tomu naopak. Zde můžeme pozorovat první rozdíl oproti srovnání pomocí ukazatele ΔSNR , ve kterém ještě pro úroveň $\theta_c = 0,3$ byla lepší metoda ClipFix.

Podle očekávání podávají nejlepší výsledky opět metody A-SPADE a CQT_rew4. Pro signál kontrabasů byla metoda CQT_rew4 dominantní, naopak pro signál z akustické kytary a orchestru vychází lépe metoda A-SPADE. V průměru však metoda CQT_rew4 podává podle hodnot PSM_t nejlepší výsledky ze všech testovaných.



Obr. 7.16: Srovnání testovaných algoritmů podle hodnot PSM_t .

Z uvedených příkladů také plyne, že ukazatel ΔSNR ne vždy koresponduje s lidským slyšením (popř. s ukazatelem PSM_t). I přes to, že podle ΔSNR byl A-SPADE zejména pro signál Vivaldi_storm.wav jednoznačně nejhorší, poslechově a také podle PSM_t podává paradoxně nejlepší výsledek.

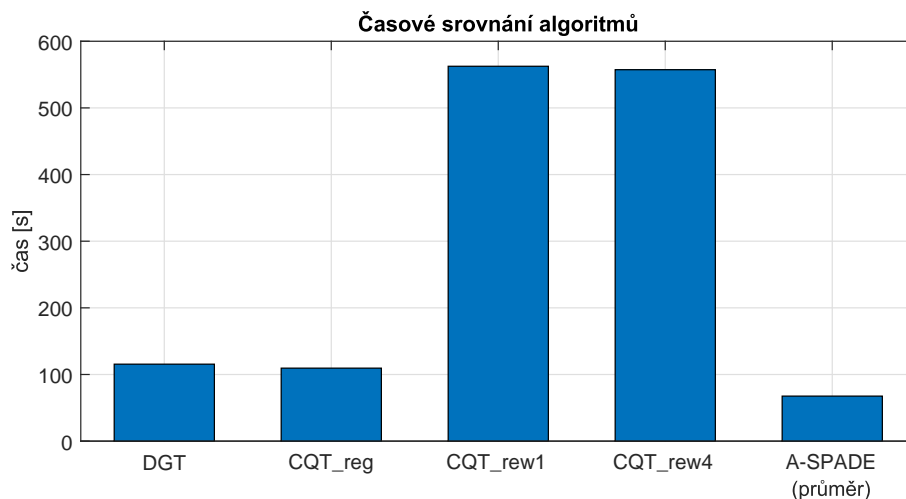
7.5 Porovnání výpočetní náročnosti algoritmů

Při navrhování představených metod byl kladen důraz především na kvalitu rekonstrukce. Dalším důležitým kritériem jednotlivých algoritmů je však také jejich výpočetní náročnost.

Porovnání výpočetní náročnosti bylo provedeno pro algoritmy s využitím slovníku DGT, neuniformní CQT, uniformní CQT s váhováním koeficientů, uniformní CQT s váhováním koeficientů a potlačením vysokofrekvenčních složek, a také pro algoritmus A-SPADE (s nastaveným profilem „accuracy“). Ukončovací kritérium algoritmů využívajících Douglas-Rachfordův algoritmus bylo 2000 iterací. Výpočetní doba těchto algoritmů byla proto stejná pro jakoukoli úroveň clippingu θ_c . Naopak algoritmus A-SPADE má ukončovací kritérium nastavené dynamicky podle relativní změny signálu v každé iteraci. Proto silněji zkreslené signály zpracovává obecně déle.

Graf 7.17 znázorňuje průměrné hodnoty výpočetních časů pro jednotlivé zvukové signály a úrovně clippingu. Je zřejmé, že nejrychlejší je algoritmus A-SPADE s průměrnou výpočetní dobou 67 sekund. Naopak nejpomalejší jsou varianty CQT, kde se provádí váhování koeficientů (CQT_rew1 a CQT_rew4). Tyto metody jsou v průměru cca pětikrát pomalejší, než metody s použitím DGT a neuniformní CQT (CQT_reg). Je to dáno jednak použitím uniformní CQT, která produkuje větší počet koeficientů signálu, a také faktem, že v případě váhování koeficientů jsou tyto koeficienty komplexní čísla a ne reálná, jako v předchozích případech.

Výpočty byly prováděny v prostředí Matlab ve verzi 2014b na PC s operačním systémem Windows 10 Pro, CPU Intel Core i5-4690K o taktu 3,5 GHz a 16 GB RAM DDR3 1333 MHz CL9.



Obr. 7.17: Srovnání výpočetní náročnosti jednotlivých algoritmů.

7.6 Poslechové testy

Posledním hlediskem porovnání algoritmů pro declipping je srovnání pomocí subjektivních poslechových testů. To probíhá pro všech pět testovaných zvukových signálů jako v případě objektivního porovnání, avšak pouze pro tři úrovně clippingu (konkrétně pro $\theta_c = 0,1; 0,3$ a $0,5$).

7.6.1 Metodika testování

Do testu jsou zahrnuty výsledky metod s použitím DGT a CQT s váhováním koeficientů a potlačením vysokofrekvenčních složek. Do srovnání jsou přidány také výsledky metod A-SPADE (s profilem „accuracy“) a ClipFix. Pro lepší srovnání je do testu zařazen také nezpracovaný saturovaný signál. Jako horní kotva poslechového testu slouží originální referenční signál, který si lze přehrát v každé fázi testování.

Kvalitu hudební rekonstrukce lze hodnotit na škále od 1 do 5 s krokem 0,5, neboli na tzv. MOS škále (*Mean Opinion Score*). Stupeň 1 znamená nejhorší kvalitu testovaného signálu a 5 naopak nejlepší možnou kvalitu. Konkrétní stupně této škály a jejich význam v případě Declippingu jsou uvedeny v tabulce 7.1.

Tab. 7.1: Pětistupňová škála pro hodnocení kvality audio signálu.

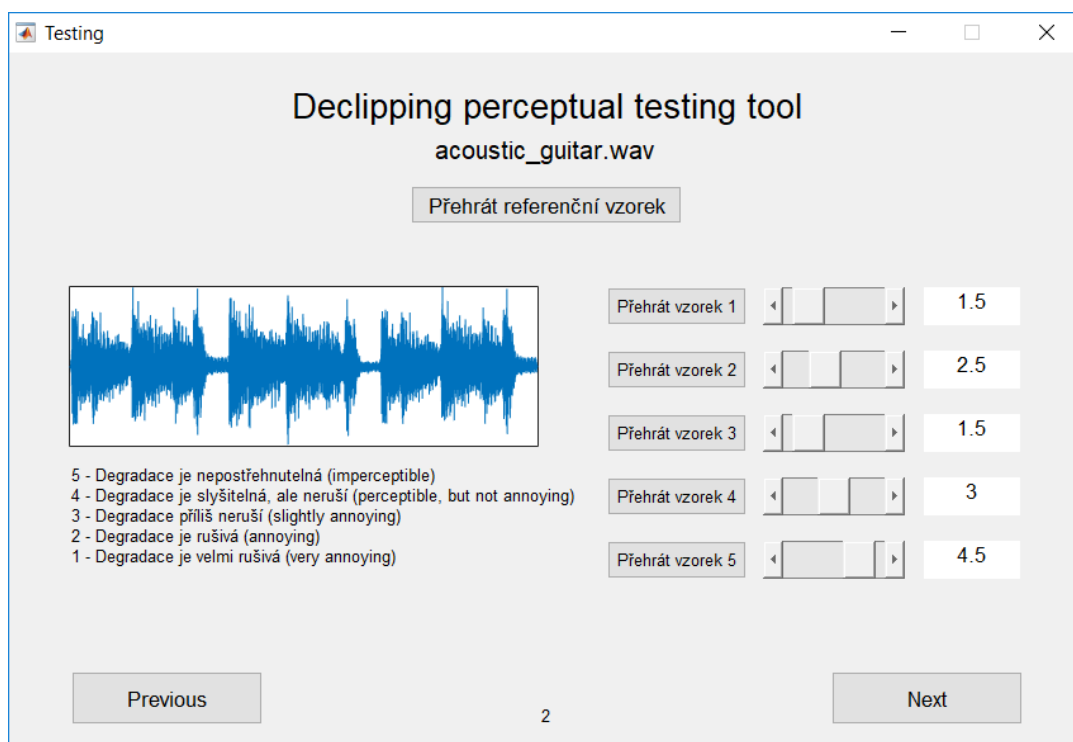
Význam snížení kvality (Impairment)	ODG
Degradace je nepostřehnutelná (imperceptible)	5,0
Degradace je slyšitelná, ale neruší (perceptible, but not annoying)	4,0
Degradace příliš neruší (slightly annoying)	3,0
Degradace je rušivá (annoying)	2,0
Degradace je velmi rušivá (very annoying)	1,0

Celkový čistý poslechový čas testu pro pět testovaných zvukových signálů a tři úrovně zkreslení pro uvedené metody činí 522 sekund (tj. 8,7 minut).

Před samotným ostrým testem jsou hodnotícímu přehrány zvukové ukázky v kvalitě, která odpovídá jednotlivým stupňům MOS škály. Tyto zvukové ukázky jsou jiné, než které se vyskytují v samotném ostrém testu. Důvodem je především připravení

hodnotícího na samotný test, což by mělo mít za následek jeho lepší orientaci na škále, a tedy i přesnější výsledky testování.

Pro rychlé a pohodlné ohodnocení poslechových ukázek byla v prostředí Matlab napsána aplikace s GUI (viz obrázek 7.18). Tato aplikace si z přiloženého souboru `sounds.mat` načte konkrétní zvukové signály a následně umožňuje pomocí tlačítek jednotlivé zvuky přehrát. Ohodnocení se nastavuje pomocí slideru vpravo od tlačítek. Po ohodnocení všech vzorků aplikace uloží výsledky do souboru typu `*.mat`.



Obr. 7.18: Screenshot aplikace pro poslechové hodnocení kvality rekonstrukce.

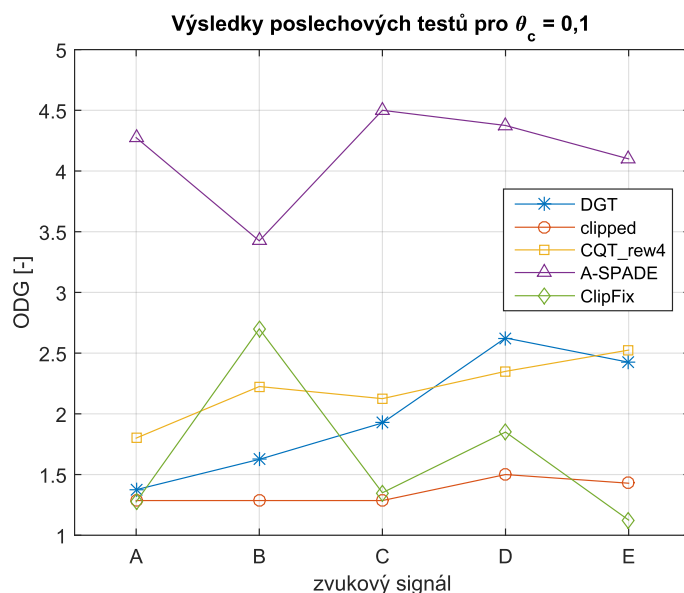
7.6.2 Výsledky poslechových testů

Poslechové testy probíhaly podle metodiky popsané v části (7.6.1). Testu se zúčastnilo celkem 20 respondentů. Výsledné grafy uvedené v této části odpovídají průměru ODG hodnot získaných od jednotlivých respondentů. Testování pro všechny respondenty proběhlo na notebooku Asus Zenbook UX32VD s externí zvukovou kartou Steinberg UR22 a se sluchátky Sennheiser HD380 Pro. Pro všechny respondenty bylo zajištěno stejné nastavení hlasitosti. Zvukové signály A, B, C, D a E označují testované signály seřazené abecedně, tedy Acoustic_guitar.wav, Double_bass.wav, Sample.wav, Speech.wav a Vivaldi_storm.wav.

Kromě výsledků testovaných metod obsahují následující grafy i výsledky saturovaného signálu. Ten je v grafech značen oranžovou barvou.

Graf 7.19 vykresluje výsledky poslechového testování pro silně zkreslené zvukové signály ($\theta_c = 0,1$). Z grafu je patrné, že podle očekávání skončil saturovaný signál s nejnižšími hodnotami ODG. Výjimku zde tvoří signál orchestru, kde metoda ClipFix získala ještě nižší skóre. Tato metoda také v rámci ODG podává nejhorší výsledky z testovaných, pouze v případě signálu kontrabasu překonává představenou metodu s využitím DGT i váhovanou CQT s potlačením vysokofrekvenčních složek.

S výjimkou řečového signálu podává metoda CQT_rew4 lepší výsledky, než metoda využívající DGT. Jako jednoznačně nejlepší vychází algoritmus A-SPADE, který s výjimkou signálu kontrabasu disponuje průměrnou hodnotou $ODG > 4$, tedy degradace je slyšitelná, ale neruší.



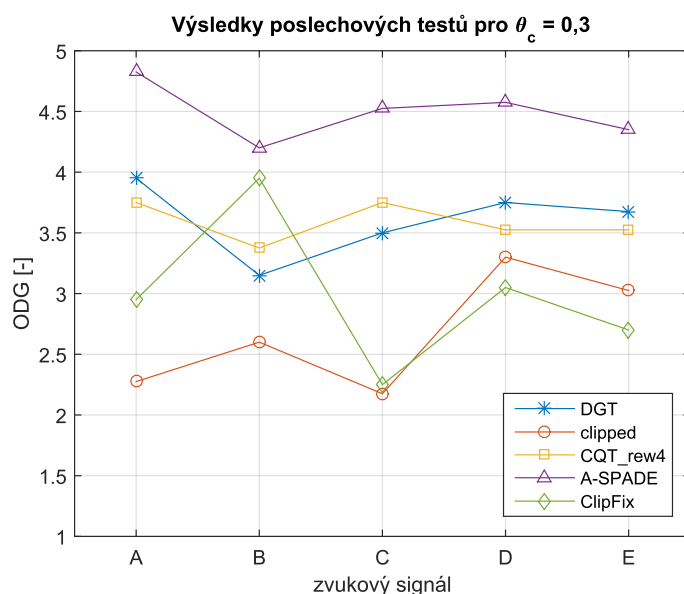
Obr. 7.19: Výsledky poslechového testování pro silně zkreslené signály ($\theta_c = 0,1$).

Na grafu 7.20 lze vidět výsledky poslechového testování pro středně zkreslené signály. V tomto případě je úroveň clippingu $\theta_c = 0,3$.

Pořadí testovaných zvukových ukázek je pro případ středního zkreslení velmi podobné jako v případě silného zkreslení. Opět vychází pro většinu testovaných signálů nejhůře saturovaný signál. Výjimku tvoří metoda ClipFix, která má nejnižší hodnoty nejen pro nahrávku orchestr, ale také pro řečový signál. Pro signál kontrabasu tato metoda však podává lepší výsledky než CQT_rew4 a DGT.

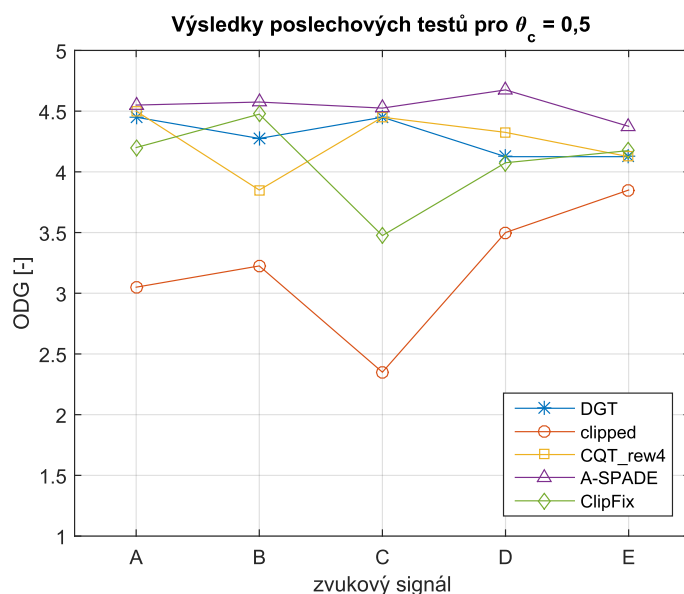
Pokud srovnáme výsledky představené metody využívající Douglas-Rachfordův algoritmus s použitím DGT a CQT s váhováním a potlačením vysokofrekvenčních koeficientů, DGT v případě středního zkreslení podává lepší výsledky u signálů akustické kytary, řeči i orchestru.

Testování pro všechny zvukové signály znovu jasně dominuje algoritmus A-SPADE.



Obr. 7.20: Výsledky poslechového testování pro středně zkreslené signály ($\theta_c = 0,3$).

V posledním případě byly testovány lehce zkreslené signály, kde $\theta_c = 0,5$. Tyto výsledky popisuje graf 7.21. Při lehkém zkreslení podle hodnot ODG vyšel jako jednoznačně nejhorší saturovaný signál. Až na metodu CQT_rew4 v případě signálu kontrabasu a metodu ClipFix v případě signálu Sample.wav, jsou ODG hodnoty rekonstruovaných signálů větší než 4, tedy zkreslení nijak neruší. I při tomto srovnání vychází jako nejlepší algoritmus A-SPADE.



Obr. 7.21: Výsledky poslechového testování pro lehce zkreslené signály ($\theta_c = 0,5$).

8 ZÁVĚR

Tato diplomová práce se věnuje problematice clippingu zvukového signálu. První kapitola se věnuje především popisu clippingu, zmiňuje dva známé druhy clippingu (hard clipping a soft clipping) a také vysvětluje, kde se s těmito jevy můžeme setkat v reálném světě.

Druhá kapitola je věnována declippingu a především jednotlivým metodám, které byly v průběhu předchozích let představeny. Celkem je zde zmíněno devatenáct metod, které jsou vždy stručně představeny, je naznačen princip, na kterém jednotlivé metody fungují, a pokud to autoři metody v článku zmiňují, jsou uvedeny i výsledky, jakých dané metody dosahují.

Cílem diplomové práce bylo také nastudovat problematiku řídké reprezentace signálů a s její pomocí formulovat metodu pro declipping audio signálu. Třetí kapitola se proto věnuje úvodu do řídké reprezentace – jsou zde objasněny základní pojmy a značení, je zmíněn aditivní model signálu, rozebrány podmínky nutné k nalezení řídkého řešení a je zde také uvedeno několik algoritmů sloužících k nalezení řídkého řešení.

Čtvrtá kapitola se zabývá teorií vektorových prostorů, bází a framů, a definuje Gaborovu transformaci a transformaci Constant-Q.

Při řešení úlohy declippingu byla nejprve provedena matematická formulace declippingu. Tento problém byl následně řešen pomocí obecného proximálního algoritmu. Tato metoda má však tu nevýhodu, že projekce probíhá zvlášť pro spolehlivé vzorky a zvlášť pro saturované vzorky. To způsobuje relativně nízkou přesnost rekonstrukce a vyšší výpočetní náročnost. Proto byly hledány způsoby jak tuto metodu dále vylepšit. Výsledkem je metoda, která používá Douglas-Rachfordův algoritmus, a která pomocí jistého triku dokáže provést projekci na množinu přípustných řešení naráz pro celý signál. Tím se zvýší přesnost rekonstrukce a sníží výpočetní náročnost. Obě tyto metody jsou popsány v kapitole Proximální algoritmy a jejich využití pro úlohu declippingu.

Všechny metody byly naprogramovány v prostředí Matlab. Jejich implementace je podrobně popsána v kapitole Programové řešení.

Kapitola 7 se zaměřuje především na metodiku testování a demonstraci jednotlivých výsledků. Nejdříve je popsána metodika testování. V další části jsou porovnány algoritmy pro declipping využívající Condatův, resp. Douglas-Rachfordův algoritmus. Z tohoto srovnání vychází lépe algoritmus Douglas-Rachford (především z pohledu výpočetní náročnosti), proto jsou v dalších částech srovnávány výsledky právě tohoto algoritmu.

Následně jsou v kapitole popsány výsledky tohoto algoritmu pro různé slovníky (konkrétně pro DGT, neuniformní CQT, uniformní CQT s váhováním koeficientů

a uniformní CQT s váhováním koeficientů a potlačením vysokofrekvenčních složek) pomocí objektivních ukazatelů ΔSNR a PEMO-Q, a také pomocí subjektivích poslechových testů. Podle obou objektivních ukazatelů se jako nejlepší z testovaných metod jeví metoda s využitím váhované uniformní CQT s potlačením vysokofrekvenčních složek. To souvisí s faktem, že většina energie zvukových signálů se nachází právě v nízkých frekvencích. Poslechové testy však ukazují, že nejkvalitnější výsledky declippingu podává stále algoritmus A-SPADE.

Ačkoli výsledky této diplomové práce podávají relativně uspokojivé výsledky, existují různé směry, kterými by se daly představené metody dále vylepšit. Nabízí se především aplikace tzv. strukturované řídkosti, při které by koeficienty nebyly vybírány jednotlivě, ale po skupinách (např. základní tón a jeho vyšší harmonické). Jako další perspektivní možnost se jeví váhovat koeficienty psychoakustickým modelem, který by mohl při rekonstrukci preferovat koeficienty, které jsou perceptuálně významnější.

LITERATURA

- [1] *Clipping (signal processing)*, Wikipedia [online]. 2016, poslední aktualizace 2.8.2016 [cit.4.11.2016]. Dostupné z URL: <[https://en.wikipedia.org/wiki/Clipping_\(signal_processing\)](https://en.wikipedia.org/wiki/Clipping_(signal_processing))>.
- [2] *Clipping (audio)*, Wikipedia [online]. 2016, poslední aktualizace 27.8.2016 [cit.4.11.2016]. Dostupné z URL: <[https://en.wikipedia.org/wiki/Clipping_\(audio\)](https://en.wikipedia.org/wiki/Clipping_(audio))>.
- [3] *Clipping (photography)*, Wikipedia [online]. 2016, poslední aktualizace 18.7.2016 [cit.4.11.2016]. Dostupné z URL: <[https://en.wikipedia.org/wiki/Clipping_\(photography\)](https://en.wikipedia.org/wiki/Clipping_(photography))>.
- [4] HARVILLA, M. J.; STERN, R. M. Efficient audio declipping using regularized least squares *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*[online]. 2015. s 221–225. Dostupné z URL: <<http://ieeexplore.ieee.org/document/7177964>>.
- [5] DEFRAENE, B.; VAN WATERSHOOT, T.; FERREAU, H. J.; DIEHL, M.; MOONEN, M. Real-Time Perception-Based Clipping of Audio Signals Using Convex Optimization. *IEEE Transactions on Audio, Speech, and Language Processing*. ročník 20, s. 2657–2671, 2012 Dostupné z URL: <<http://ieeexplore.ieee.org/document/6255767/>>.
- [6] DEFRAENE, B.; MANSOUR, N.; DE HERTOOGH, S.; VAN WATERSCHOOT, T.; DIEHL, M.; MOONEN M. Declipping of Audio Signals Using Perceptual Compressed Sensing. *IEEE Transactions on Audio, Speech, and Language Processing* [online]. 2016. s 2627–2637. Dostupné z URL: <<http://ieeexplore.ieee.org/document/6600777>>.
- [7] JANSSEN, A.; VELDHUIS, R.; VRIES, L. Adaptive interpolation of discrete-time signals that can be modeled as autoregressive processes. *IEEE Transactions on Acoustics, Speech and Signal Processing* [online]. 1986. s 317–330. Dostupné z URL: <<http://ieeexplore.ieee.org/document/1164824/>>.
- [8] ABEL, J. S.; SMITH, J.O. Restoring a clipped signal. *Proc. Int. Conf. Acoustics, Speech and Signal Processing(ICASSP)*. vydání 3, 1991. s. 1745–1748. Dostupné z URL: <<http://ieeexplore.ieee.org/document/150655/>>.
- [9] GODSILL, S. J.; WOLFE, P. J.; FONG, W. N. Statistical model-based approaches to audio restoration and analysis. *J. New Music Res.* vydání 30, 2001, s. 323–338.

- [10] FONG, W.; GODSILL, S. Monte Carlo smoothing for non-linearly distorted signals. *IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings*. vydání 6, 2001, s. 3997–4000. Dostupné z URL: <<http://ieeexplore.ieee.org/document/940720/>>.
- [11] OLOFSSON, T.; Deconvolution and model-based restoration of clipped ultrasonic signals. *IEEE Transactions on Instrumentation and Measurement*, vol. 54, no. 3, s 1235–1240, 2005. Dostupné z URL: <<http://ieeexplore.ieee.org/document/1433200/>>.
- [12] DAHIMENE, A.; NOUREDDINE, M.; AZRAR, A. A Simple Algorithm for the Restoration of Clipped Speech Signal. *Informatica* ročník 32, s. 183–188, 2008. Dostupné z URL: <https://www.researchgate.net/publication/220166548_A_Simple_Algorithm_for_the_Restoration_of_Clipped_Speech_Signal>.
- [13] MIURA, S.; NAKAJIMA, H.; MIYABE, S.; MAKINO, S.; YAMANDA, T.; NAKADAI, K. Restoration of Clipped Audio Signal Using Recursive Vector Projection. *TENCON 2011 - 2011 IEEE Region 10 Conference 2011*, s. 394–397. Dostupné z URL: <<http://ieeexplore.ieee.org/document/6129132/>>.
- [14] ADLER, A.; EMIYA, V.; JAFARI, M. G.; ELAD, M.; GRIBONVAL, R.; PLUMBLEY, M. D. A constrained matching pursuit approach to audio declipping. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) 2011*, s. 329–332. Dostupné z URL: <<http://ieeexplore.ieee.org/document/5946407/>>.
- [15] WEINSTEIN, A. J.; WAKIN, M. B. Recovering a Clipped Signal in Sparseland *Sampling Theory in Signal and Image Processing*. ročník 12, s. 55–69, 2013.
- [16] SALESNICK, I. Least Squares with Examples in Signal Processing.[online] OpenStax CNX. 28. 4. 2013, [cit. 5. 4. 2017]. Dostupné z URL: <<https://cnx.org/contents/XRPKcVgh@1/Least-Squares-with-Examples-in>>.
- [17] KITIĆ, S.; JACQUES, L.; MADHU, N.; HOPWOOD, M. P.; SPRIET, A.; DE VLEESCHOUWER, C. Consistent Iterative Hard Thresholding for Signal Declipping. 2013. Dostupné z URL: <https://www.researchgate.net/publication/235772084_Consistent_Iterative_Hard_Thresholding_For_Signal_Declipping>.

- [18] HRBÁČEK, R.; RAJMIC, P.; VESELÝ, V.; ŠPIŘÍK, J. *Řídké reprezentace signálu: komprimované snímání* [online]. Elektrevue, 2011. ISSN 1213-1539. Dostupné z URL: <<http://elektrevue.cz/cz/download/ridke-reprezentace-signalu--komprimovane-snimani/>>.
- [19] BS.1387: Method for objective measurements of perceived audio quality. *ITU (International Telecommunication Union)* [online]. 2001, poslední aktualizace 10.6.2015 [cit. 21.1.2017]. Dostupné z URL: <<http://www.itu.int/rec/R-REC-BS.1387-1-200111-I/en>>.
- [20] SIEDENBURG, K.; KOWALSKI, M.; DÖRFLER, M. Audio Declipping With Social Sparsity. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2014, s. 1577–1581 Dostupné z URL: <<http://ieeexplore.ieee.org/document/6853863/>>.
- [21] Audio Declipping with Social Sparsity. *Structured Sparsity for Audio Signals*, NuHAG [online]. poslední aktualizace 7.3.2016 [cit. 21.1.2017]. Dostupné z URL: <<http://homepage.univie.ac.at/monika.doerfler/StrucAudio.html#Declipping>>.
- [22] KITIĆ, S.; BERTIN, N.; GRIBONVAL, R. Audio Declipping by Cosparsity Hard Thresholding. *iTwist – Traveling Workshop on Interactions between Sparse models and Technology*. 2014. Dostupné z URL: <<https://hal.inria.fr/hal-00922497>>.
- [23] JONSCHER, M.; SEILER, J.; KAUP, A. Declipping of Speech Signals Using Frequency Selective Extrapolation. *ITG-Fachbericht 252: Speech Communication* 2014, s. 1–4. Dostupné z URL: <<http://ieeexplore.ieee.org/document/6926053/>>.
- [24] HARVILLA, M. J.; STERN, R. M. Least Squares Signal Declipping for Robust Speech Recognition. *INTERSPEECH* 2014. Dostupné z URL: <<https://pdfs.semanticscholar.org/7c11/8586b5d602481ccfe4556ec0e8b5d0057942.pdf>>.
- [25] HARVILLA, M. J.; STERN, R. M. Efficient Audio Declipping Using Regularized Least Squares. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* 2015, s. 221–225. Dostupné z URL: <<http://ieeexplore.ieee.org/document/7177964/>>.

- [26] BILEN, Ç; OZEROV, A; PÉREZ, P. Audio declipping via nonnegative matrix factorization. *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. 2015, s. 1–5. Dostupné z URL: <<http://ieeexplore.ieee.org/document/7336948/>>.
- [27] KITIĆ, S.; BERTIN, N.; GRINBONVAL, R. Sparsity and Cosparsity for Audio Declipping: A Flexible Non-convex Approach. *LVA/ICA 2015, LNCS 9237* s. 243–250, 2015. Dostupné z URL: <<https://arxiv.org/pdf/1506.01830v2.pdf>>.
- [28] Software. *Srdan Kitić* [online]. 2016, [cit. 9. 12. 2016]. Dostupné z URL: <https://people.rennes.inria.fr/Srdan.Kitic/?page_id=40>.
- [29] HRBÁČEK, R.; RAJMIC, P.; VESELÝ, V.; ŠPIŘÍK, J. *Řídká reprezentace signálů: úvod do problematiky* [online]. Elektrorevue, 2011. ISSN 1213-1539. Dostupné z URL: <<http://www.elektrorevue.cz/cz/download/ridke-reprezentace-signalu--uvod-do-problematiky/.e.cz/>>.
- [30] NAM, S.; DAVIES, M. E.; ELAD, M.; GRIBONVAL, R. *The Cosparsity Analysis Model and Algorithms* [online]. 2011, 41 s. Dostupné z URL: <<https://arxiv.org/pdf/1106.4987v1.pdf>>.
- [31] COHEN, A.; DAHMEN, W.; DEVORE, R. Compressed sensing and best term approximation. *Journal of The American Mathematical Society*, ročník 22, 2009. s 211–231.
- [32] CANDÈS, E. J.; TAO, T.: Decoding by Linear Programming. *IEEE Transactions on Information Theory*, ročník 51, 2005. s 4203–4215.
- [33] RAJMIC, P. *Řídké a nízkohodnostní reprezentace signálů s aplikacemi*. Brno, 2014. Habilitační práce. VUT v Brně.
- [34] BROWN, J., C.; PUCKETTE, M., S. An efficient algorithm for the calculation of a constant Q transform. *J. Acoust. Soc. Am.* 1992, s. 2698–2701. Dostupné z URL: <<http://academics.wellesley.edu/Physics/brown/pubs/effalgV92P2698-P2701.pdf>>.
- [35] CQTFILTERS - CQT-spaced filters. *LTFAT – The Large Time-Frequency Analysis Toolbox* [cit. 2. 3. 2017]. Dostupné z URL: <<http://ltfat.github.io/doc/filterbank/cqtfilters.html>>.

- [36] HOLIGHAUS, N.; DÖRFLER, M.; VELASCO, G. A.; GRILL, T. A Framework for Invertible, Real-Time COnstant-Q Transforms. *IEEE Transactions on Audio, Speech, and Language Processing* 2012, s. 775–785. Dostupné z URL: <<http://ieeexplore.ieee.org/document/6384709/>>.
- [37] COMBETTES, L.P.; PESQUET, J-C. *Proximal splitting methods in signal processing*. Fixed-Point Algorithms for Inverse Problems in Science and Engineering, 2011: 28 s. Dostupné z URL: <<http://arxiv.org/abs/0912.3522v4>>.
- [38] CONDAT, Laurent. A Generic Proximal Algorithm for Convex Optimization – Application to Total Variation Minimization *IEEE Signal Processing Letters*, ročník 21, 2014. s. 985–989. Dostupné z URL: <<https://www.gipsa-lab.grenoble-inp.fr/~laurent.condat/publis/Condat-optim-SPL-2014.pdf>>.
- [39] CONDAT, Laurent. A primal-dual splitting method for convex optimization involving Lipschitzian, proximable and linear composite terms. *Journal of Optimization Theory and Applications*, ročník 158, vydání 2, 2013. s. 460–479.
- [40] RAJMIC, P. Projection onto set of linear equations defined by reduced dictionary. 2017, poslední aktualizace 24.3.2017, [cit.30.3.2017]. Dostupné z URL: <http://www.utko.feec.vutbr.cz/~rajmic/papers/projection_reduced_dictionary.pdf>.
- [41] RAJMIC, P.; VESELÝ, V. Projection onto a multidimensional interval with a particular class of full-rank linear operators. 2017, poslední aktualizace 24.3.2017, [cit.30.3.2017]. Dostupné z URL: <http://www.utko.feec.vutbr.cz/~rajmic/papers/projection_frame_interval.pdf>.
- [42] PRŮŠA, Z.; SØNDERGAARD, P.; HOLIGHAUS, N.; WIESMEYR, CH.; BALAZS, P. The Large Time-Frequency Analysis Toolbox 2.0. *Sound, Music and Motion, Lecture Notes in Computer Science* 2014, s 419–442. Dostupné z URL: <<http://lthfat.github.io/notes/lthfatnote030.pdf>>.
- [43] Clip Fix [online]. 2017, [cit.20.5.2017]. Dostupné z URL: <http://manual.audacityteam.org/man/clip_fix.html>.
- [44] HUBER, R.; KOLLMEIER, B. PEMO-Q: A New Method for Objective Audio Quality Assessment Using a Model of Auditory Perception. *IEEE Transactions on Audio, Speech, and Language Processing* 2006, s 1902–1911. Dostupné z URL: <<http://ieeexplore.ieee.org/document/1709880/>>.
- [45] PEMO-Q - HörTech gGmbH [online]. 2017, [cit.20.5.2017]. Dostupné z URL: <<http://www.hoertech.de/en/f-e-products/pemo-q.html>>.

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

A-SPADE	Analysis Sparse Audio Declipper
BP	Basis Pursuit
C-IHT	Consistent Iterative Hard Thresholding
C-OMP	Constrained Orthogonal Matching Pursuit
CBAR	Constrained Blind Amplitude Reconstruction
CCD	Charged-Coupled Device – zařízení pro pohyb elektrického náboje, převážně využíváno jako světelný senzor v digitálních fotoaparátech
CMOS	Complementary Metal-Oxide-Semiconductor – technologie pro výrobu integrovaných obvodů
CoDecHT	Cospase Declipping by Hard Thresholding
CQT	Constant-Q non-stationary Gabor filterbank
CSL0	Compressed Sensing based method using ℓ_0 norm
CSL1	Compressed Sensing based method using ℓ_1 norm
DCT	Discrete Cosine Transform – Diskrétní kosinová transformace
DFT	Discrete Fourier Transform – Diskrétní Fourierova transformace
DGT	Discrete Gabor Transform – Diskrétní Gaborova transformace
EW	Empirical Weiner
FSE	Frequency Selective Extrapolation – Frekvenčně selektivní extrapolace
GUI	Graphical User Interface – grafické uživatelské rozhraní
IDFT	Inverse Discrete Fourier Transform – inverzní diskrétní Fourierova transformace
IRLS	Iterative Reweight Least Squares
ISTA	Iterative Shrinkage/Thresholding Algorithm
LARS	Least Angle Regression
MOS	Mean Opinion Score – škála od 1 do 5 pro ohodnocení kvality audia a videa
MP	Matching Pursuit
NMF	Non-negative Matrix Factorization – faktorizace nezáporných matic
NSP	Null Space Property – Vlastnost nulového prostoru
OMP	Orthogonal Matching Pursuit
OLA	Overlap-add – metoda přičítání přesahu při zpracovávání signálu po blocích
PCSL1	Perceptual Compressed Sensing based method using ℓ_1 norm
PEAQ ODG	Perceptual Evaluation of Audio Quality Objective Difference Grade – ukazatel objektivně hodnotící kvalitu zvukového signálu podle subjektivního lidského vnímání

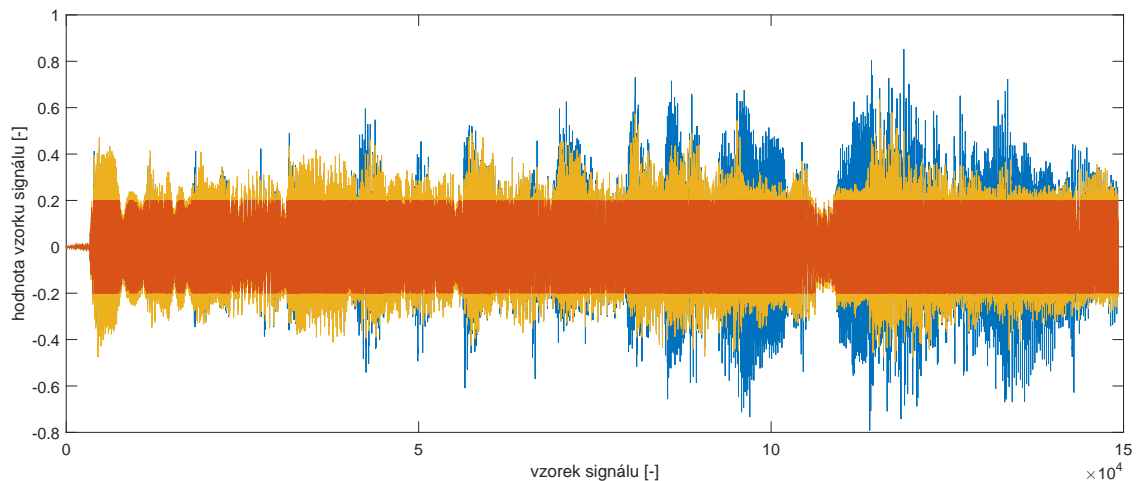
PEW	Persistent Empirical Weiner
PSM	Perceptual Similarity Measure – ukazatel hodnotící podobnost zvukového signálu na základě subjektivního lidského vnímání
RBAR	Regularized Blind Amplitude Reconstruction
RIP	Restricted Isometry Property – Vlastnost zeslabené isometrie
$R\ell_1CC$	Reweighted ℓ_1 minimization with Clipping Constraints
S-SPADE	Synthesis Sparse Audio Declipper
SNR	Signal to Noise Ratio – Odstup signálu od šumu
STFT	Short Time Fourier Transform – krátkodobá Fourierova transformace
TPCC	Trivial Pursuit with Clipping Constraints
WER	Word Error Rate
WGL	Windowed Group-Lasso

SEZNAM PŘÍLOH

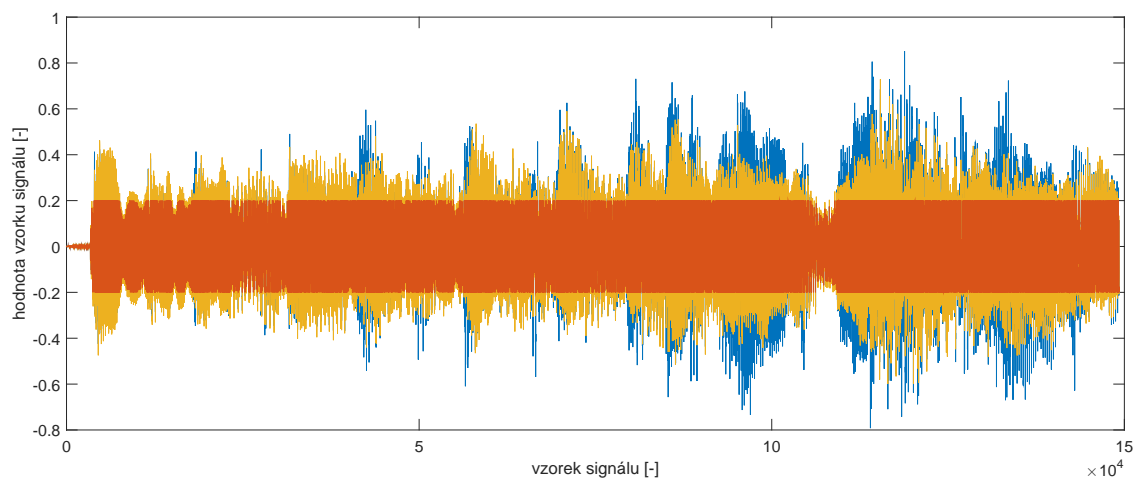
A	Grafy rekonstruovaných signálů	91
B	Obsah přiloženého CD	93
B.1	Sparse_Declip	93
B.2	Sounds	93
B.3	Testing	94

A GRAFY REKONSTRUOVANÝCH SIGNÁLŮ

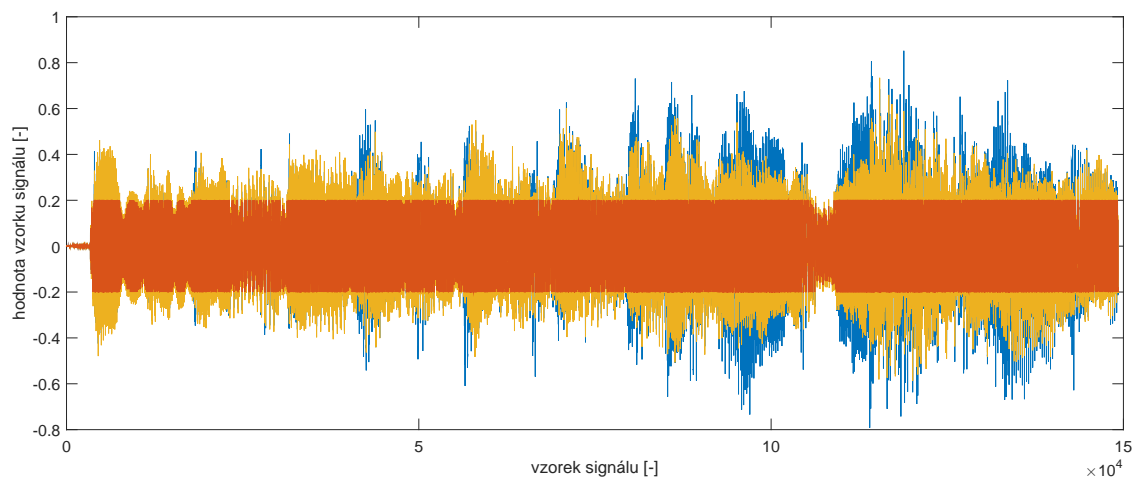
V této příloze budou pro demonstraci představených metod zobrazeny grafy rekonstruovaného signálu Sample.wav v časové oblasti pro úroveň clippingu $\theta_c = 0,2$. Modrou barvou je v grafech vykreslen referenční signál, oranžovou saturovaný signál a žlutou barvou rekonstruovaný signál.



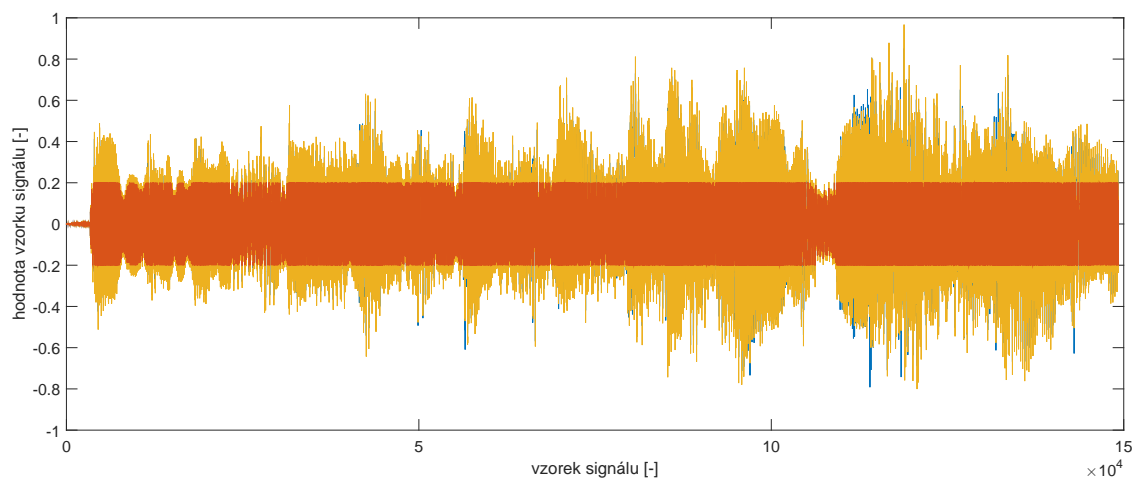
Obr. A.1: Ukázka rekonstrukce signálu se slovníkem DGT.



Obr. A.2: Ukázka rekonstrukce signálu s použitím neuniformní CQT.



Obr. A.3: Ukázka rekonstrukce signálu s použitím váhované uniformní CQT.



Obr. A.4: Ukázka rekonstrukce signálu s použitím váhované uniformní CQT a potlačením vysokofrekvenčních složek.

B OBSAH PŘÍLOŽENÉHO CD

Na přiloženém CD lze najít zdrojové kódy naprogramovaných metod, řešících declipping zvukových signálů. Dále jsou zde přiloženy zvukové signály ve formátu *.wav, a to nejen nezpracované originální signály, ale i rekonstruované. Dále je zde pak obsažena aplikace s GUI určená pro poslechové testy jednotlivých metod. Pro přehlednost je obsah CD rozdělen do následujících adresářů.

B.1 Sparse_Declip

V tomto adresáři se nacházejí zdrojové kódy balíku pro declipping pomocí řídkých reprezentací signálů. Uvedené soubory jsou podobně popsány v kapitole 6.

1. `condat.m` – implementace Condatova algoritmu pro potřeby declippingu (viz část 6.4)
2. `cqttightframe.m` – funkce, která generuje parsevalovsky těsný frame CQT banky filtrů a vektor vah koeficientů (autor Zdeněk Průša) (viz část 6.1)
3. `declipping_main.m` – hlavní soubor balíku declipping (viz část 6.1)
4. `doug_rach.m` – implementace Douglas-Rachfordova algoritmu pro potřeby declippingu (viz část 6.5)
5. `hard_clip.m` – funkce pro tvrdé ořezání vstupního signálu (viz část 6.3)
6. `proj_box.m` – funkce pro výpočet projekce na multidimenzionální prostor (autor Pavel Rajmic), (viz část 6.7)
7. `proj_parse_frame.m` – funkce pro výpočet projekce v případě declippingu pomocí Douglas-Rachfordova algoritmu (viz část 6.7)
8. `segmentation.m` – funkce pro zpracování vstupního signálu po blocích (viz část 6.6)
9. `snr_n.m` – funkce pro počítání hodnot SNR vstupních signálů (viz část 7.3)
10. `soft_clip.m` – funkce pro měkké ořezání vstupního signálu (viz část 6.3)
11. `soft_clip2.m` – funkce pro měkké ořezání vstupního signálu (viz část 6.3)
12. `win_param.m` – funkce pro výpočet parametrů okna pro DGT (viz část 6.2)

B.2 Sounds

V tomto adresáři se nacházejí všechny zvukové signály. Je rozdělen do několika podadresářů, které rozlišují algoritmus, pomocí kterého byly zvuky zpracovány. Uvnitř těchto podadresářů je dalších 9 podadresářů (01 až 09), které označují úroveň clipingu θ_c zpracovaných zvukových signálů.

1. **A-SPADE accuracy** – signály rekonstruované algoritmem A-SPADE s nastaveným profilem accuracy.
2. **A-SPADE fast** – signály rekonstruované algoritmem A-SPADE s nastaveným profilem fast.
3. **audacity_clipFix** – signály rekonstruované metodou clipFix v editoru Audacity
4. **clipped** – tvrdě omezené (saturované) signály
5. **Declip_CA_cqt** – signály rekonstruované pomocí Condatova algoritmu s použitím neuniformní CQT
6. **Declip_CA_cqt_uniform** – signály rekonstruované pomocí Condatova algoritmu s použitím uniformní CQT s váhováním koeficientů
7. **Declip_CA_dgtreal** – signály rekonstruované pomocí Condatova algoritmu s použitím DGT
8. **Declip_DR_cqt** – signály rekonstruované pomocí Douglas-Rachfordova algoritmu s použitím neuniformní CQT
9. **Declip_DR_cqt_uniform** – signály rekonstruované pomocí Douglas-Rachfordova algoritmu s použitím uniformní CQT s váhováním koeficientů
10. **Declip_DR_cqt_uniform_rew4** – signály rekonstruované pomocí Douglas-Rachfordova algoritmu s použitím uniformní CQT s váhováním koeficientů a potlačením vysokofrekvenčních složek
11. **Declip_DR_dgtreal** – signály rekonstruované pomocí Douglas-Rachfordova algoritmu s použitím DGT
12. **original** – původní „čisté“ signály

B.3 Testing

V tomto adresáři se nacházejí zdrojové kódy a potřebné soubory pro spuštění aplikace pro poslechové testování kvality zvukových signálů.

1. **Testing.fig** – soubor obsahující informace o prvcích GUI aplikace Declipping perceptual testing tool
2. **Testing.m** – soubor obsahující vlastní implementaci aplikace Declipping perceptual testing tool
3. **sounds.mat** – soubor obsahující zvukové signály pro testování, které se načítají při inicializaci programu
4. **picture.png** – obrázek rekonstruovaného signálu sloužící pro titulní stranu programu